Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

FEATuRE - Ferramenta para a geração da representação *Bag-of-related-words*

Rafael Geraldeli Rossi
Solange Oliveira Rezende

Nº XXX

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Agosto/2011

Resumo

Neste relatório técnico é apresentada a ferramenta FEATuRE, utilizada para a geração da representação *bag-of-related-words*. A ferramenta disponibiliza os quatro passos definidos para a geração da representação *bag-of-related-words*: i) mapear um documento textual em transações; ii) extrair regras de associação das transações mapeadas de um documento; iii) utilizar os itens das regras para compor os atributos de um documento; e iv) construir uma representação baseada no modelo espaço vetorial utilizando os atributos gerados. Além disso, a ferramenta apresenta funcionalidades para apoio à geração da representação *bag-of-related-words*, como a visualização o número de regras geradas para aumentar ou diminuir os parâmetros definidos para a geração das regras, e a análise da representação gerada, como a comparação de atributos de diferentes representações, contagem do número de atributos compostos por n palavras e conversão da representação *bag-of-related-words* gerada para o formato `arff`. Este relatório técnico apresenta detalhes sobre as telas, as entradas e saídas dos passos e as iterações entre as principais classes da ferramenta.

Palavras-chave: Mineração de Textos, Representação de Coleções de Documentos Textuais.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A quantidade de dados disponível em formato digital na rede mundial de computadores tem aumentado incessantemente. De acordo com estimativas realizadas em 2010, somente em 2009 a quantidade de dados armazenada chegou à aproximadamente 800 mil petabytes. A previsão para 2020 é que a quantidade de dados criada e replicada no mundo deve chegar a 35 milhões de petabytes (Gantz e Reinsel, 2010).

Parte dos dados no universo digital estão no formato textual, como *e-mails*, relatórios, boletins, artigos e conteúdo de páginas de internet. Tais dados podem ter uma grande quantidade de conhecimento embutido. Entretanto, é humanamente impossível analisar, organizar e extrair conhecimento útil de toda essa grande quantidade de dados textuais disponível de forma manual. Consequentemente, o uso de técnicas automáticas para organizar, gerenciar e descobrir conhecimento útil em textos está se tornando cada vez mais importante. Para auxiliar o usuário nessa tarefa, técnicas de Mineração de Textos são fundamentais (Gupta e Lehal, 2009; Feldman e Sanger, 2006).

Para que dados textuais brutos possam tornar-se úteis, é necessário que estes sejam representados de maneira apropriada para a manipulação dos algoritmos usados na Mineração de Textos. O modelo espaço vetorial (Salton, 1989) é tipicamente usado na representação de documentos textuais (Song e fang Brook Wu, 2008; Feldman e Sanger, 2006; Ebecken et al., 2003). Neste modelo, cada documento é representado por um vetor, e cada posição desse vetor corresponde a uma dimensão (atributo/termo) da coleção de documentos. Esses atributos geralmente são palavras simples, conjuntos de palavras, ou frases. O valor atribuído a cada dimensão pode indicar a ausência/presença do atributo no documento (0 ou 1), a frequência do atributo no documento, entre outras. O modelo espaço vetorial é simples e permite o uso dos métodos tradicionais que lidam com vetores de atributos numéricos, além de ser construído e computado facilmente (Schenker et al., 2003).

Uma abordagem baseada no modelo espaço vetorial e uma das mais utilizadas na Mineração de Textos é a representação *bag-of-words*. Nesta abordagem extremamente simples, cada palavra encontrada na coleção de documentos pode tornar-se uma dimensão no espaço vetorial. Porém, são ignoradas a ordem das palavras ou as palavras ao redor, informação de pontuação ou estrutural, e assume-se que a ocorrência das palavras em um documento são independentes, suposição que não é válida para documentos textuais reais. Este tipo de representação apresenta alta dimensionalidade mesmo após a etapa de pré-processamento. Devido à alta dimensionalidade, muitos atributos apresentam valor zero para a frequência em um documento, gerando assim um desperdício de espaço/memória. Muitos algoritmos são computacionalmente ineficientes ao lidar com matrizes esparsas e de alta dimensionalidade.

Outro problema encontrado ao utilizar palavras simples como atributos dos documentos textuais é que conceitos do mundo real muitas vezes são compostos por duas ou mais palavras, e gerar atributos compostos por mais de uma palavra pode ser útil para o processo de Mineração de Textos. Um exemplo retirado de (Scott e Matwin, 1999) ilustra bem esta situação. Considere a frase "*machine learning*". Esta frase tem um significado específico que é distinto do significado das palavras "*machine*" e "*learning*". Considere também uma situação em que se utiliza um classificador baseado em regras, e as classes a serem aprendidas são "*Artificial Intelligence*", "*Machine Tools*" e "*Education*". É concebível que a frase "*machine learning*" produz um alto ganho de informação para a classe "*Artificial Intelligence*", mesmo que cada palavra individual produza um baixo ganho, pois a palavra "*machine*" pode ocorrer tanto nas classes "*Artificial Intelligence*" e "*Machine Tools*", e a palavra "*learning*" pode ocorrer tanto nas classes "*Artificial Intelligence*" e "*Education*". Neste caso, uma regra potencialmente útil como "*machine*" & "*learning*" $\rightarrow$ "*Artificial Intelligence*" poderia nunca ser aprendida.

A utilização de atributos compostos por mais de uma palavra também pode tornar o conhecimento extraído ou a identificação de documentos mais interpretável ao usuário. Para ilustrar isso, vamos considerar um caso em que se deseja gerar descritores para um documento textual referente à medidas de validação para agrupamentos utilizando o algoritmo K-means (Wu et al., 2009). Ao considerar as 10 palavras simples mais frequentes do documento, são extraídos os seguintes descritores: *measure*, *cluster*, *validation*, *data*, *class*, *normal*, *k-means*, *results*, *sets*, *distribution*. Ao utilizar conjuntos de duas palavras mais frequentes são extraídos os seguintes descritores: *cluster_measure*, *validation_measures*, *data_sets*, *measure_normalization*, *cluster_validation*, *k-means_clustering*, *clustering_results*, *results_k-means*, *measure_k-means*, *external_measures*, e *data_clustering*. Pode-se notar que os conjuntos de duas palavras mais frequentes facilitam mais o entendimento do assunto do documento do que as palavras simples mais frequentes.

Essas observações levaram os pesquisadores a desenvolver alternativas para representação de documentos textuais mais ricas que a representação *bag-of-words*, como as representações baseadas em frases (n-gramas) (Carvalho et al., 2010; Carvalho e Cohen, 2006; Fürnkranz, 1998; Mladenic e Grobelnik, 1998; Fagan, 1989) e em conjuntos de palavras (Zhang e Zhu, 2007; Tesar et al., 2006; Bekkerman e Allan, 2004; Yang et al., 2003; Ahonen-Myka et al., 1999). Essas abordagens geralmente adicionam atributos à representação *bag-of-words*, aumentando ainda mais a dimensionalidade da representação. Algumas abordagens analisam a coleção inteira para gerar os atributos. Isto pode aumentar o custo computacional devido a grande quantidade de textos a ser analisada. Além disso, a maioria das abordagens necessita executar um processo de seleção de atributos supervisionado, requerendo assim coleções textuais rotuladas, o que não é comum em coleções de documentos textuais reais. Portanto, um algoritmo eficiente para gerar esses conjuntos de atributos deve então evitar a geração de uma grande quantidade de atributos para evitar o processo de seleção de atributos (Fürnkranz, 1998).

Para evitar estes problemas apresentados ao gerar atributos compostos por mais de uma palavra, em (Rossi e Rezende, 2011b,a) é apresentada a representação *bag-of-related-words*. Esta representação tem como objetivo gerar atributos compostos por palavras relacionadas analisando cada documento individualmente da coleção, de forma que o número de atributos seja menor que a da representação *bag-of-words*, com padrões extraídos de qualidade tão boa quanto e com resultados de mais fácil entendimento.

Para gerar a representação *bag-of-related-words* foi desenvolvida a ferramenta FEATuRE (Features gEnerator based on AssociaTion RulEs). Essa ferramenta viabiliza os 4 passos para a geração da representação *bag-of-related-words* propostos em Rossi (2011) e uma série de funcionalidades para a análise e conversão para outros formatos.

O restante deste relatório técnico está dividido da seguinte forma. No Capítulo 2 são apresentados os detalhes da representação *bag-of-related-words*. No Capítulo 3 são apresentadas as funcionalidades da a ferramenta FEATuRE, desenvolvida para implementar os passos para geração da representação *bag-of-words*. Por fim, no Capítulo 4 são apresentadas as considerações finais sobre este relatório.

Capítulo 2

Bag-of-Related-Words

O objetivo da representação *bag-of-related-words* é utilizar como atributos palavras relacionadas que se repetem, em espaços limitados, ao longo de um documento (Rossi e Rezende, 2011b,a). Esses atributos, além de auxiliarem a extração de padrões, são mais significativos aos usuários do processo de Mineração de Textos.

As palavras relacionadas são obtidas por meio de regras de associação (Agrawal e Srikant, 1994). Uma regra de associação é uma regra do tipo $A \Rightarrow B$, na qual A e B são grupos de itens, chamados de *itemsets*, e $A \cup B = \emptyset$. As regras de associação extraem relações entre items em uma base de dados, na qual $A \Rightarrow B$ significa que quando A ocorre, B também tende a ocorrer. Duas medidas clássicas para gerar regras de associação são `suporte` e `confiança`. O suporte mede a probabilidade conjunta dos *itemsets* em A e B ocorrerem na base de dados, isto é, $sup(A \Rightarrow B) = n(A \cup B)/N$, na qual $n(A \cup B)$ é o número de transações em que os *itemsets* de A e B ocorrem juntos, e N é o número total de transações. A confiança indica a probabilidade dos *itemsets* de A e B ocorrerem juntos dado que A ocorreu, isto é, $conf(A \Rightarrow B) = n(A \cup B)/n(A)$. Usualmente valores mínimos de suporte e confiança são definidos para gerar as regras.

Os quatro passos principais para a construção da representação *bag-of-related-words* são:

1. Mapear o documento textual em transações;
2. Extrair regras de associação das transações;
3. Utilizar os itens das regras para compor os atributos;
4. Utilizar os atributos para construir a matriz atributo-valor;

Uma ilustração geral desses 4 passos é apresentado na Figura 2.1. Nas próximas seções serão detalhados os passos para a geração da representação *bag-of-related-words*.

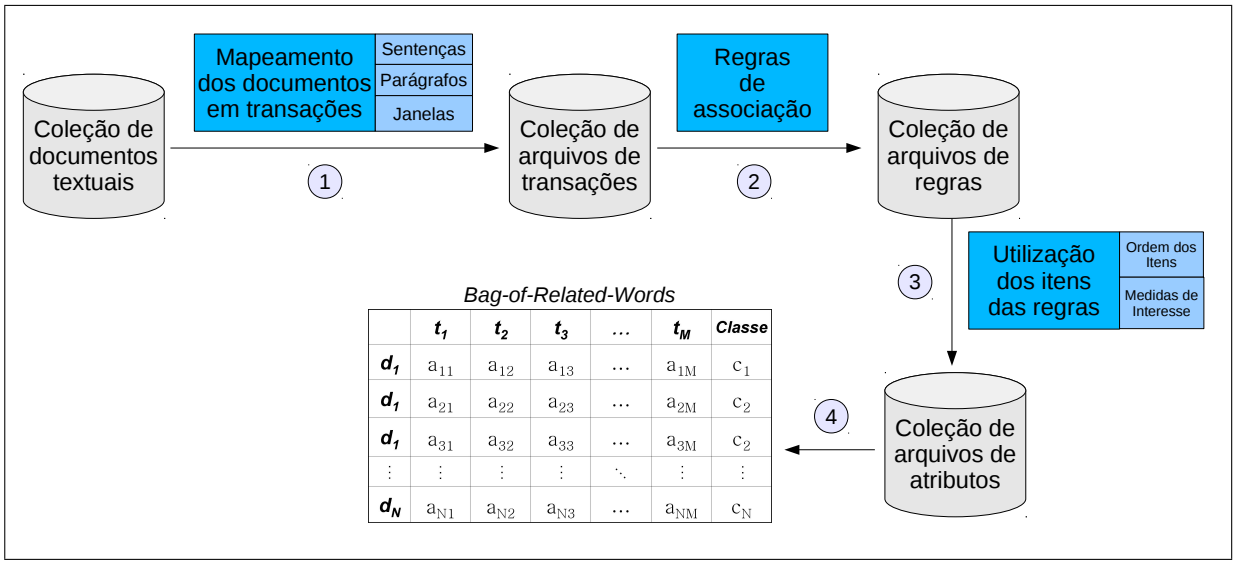


Figura 2.1: Passos para a geração da representação *bag-of-related-words*.

2.1 Mapeando um Documento Textual em Transações

Para que se possam extrair as regras de associação de um único documento, é necessário mapear esse documento em um conjunto de transações. Neste trabalho foram avaliadas três formas de realizar este mapeamento: utilizando as sentenças do documento como transações, utilizando parágrafos como transações, e utilizando janelas deslizantes como transações. Como as transações correspondem à "pedaços" do documento textual, intuitivamente serão extraídos conjuntos de palavras que estão relacionados em um contexto específico do documento, e não conjuntos de palavras que aparecem espalhadas em um mesmo documento ou em vários documentos de uma coleção.

Foram consideradas como sentenças neste projeto, todas as palavras entre os sinais ".", "!", ou "?". Como parágrafos, são consideradas as palavras entre os sinais de pontuação ".", "!", ou "?" que são seguidos por uma quebra de linha.

No mapeamento de janelas deslizantes, a primeira transação contém apenas a primeira palavra do documento, a segunda contém as duas primeiras palavras, e assim por diante, até que a janela contenha o número de palavras igual ao tamanho definido para a janela (d)[1]. Após isso, a janela desliza uma palavra e considera as próximas d palavras do documento. Considerando que o documento tenha sido mapeado em l transações, a transação $l - d$ irá conter as d últimas palavras, a transação $l - d + 1$ as $d - 1$ palavras, e assim por diante, até que a última transação contenha apenas uma palavra. O mapeamento de janelas deslizantes foi definido desta forma para que todas as palavras do texto estejam

[1]As janelas deslizantes com seus respectivos tamanhos serão expressos por "janela *Tamanho*". Por exemplo, uma janela deslizante de tamanho 5 será expressa por "janela 5".

contidas o mesmo número de vezes na janela deslizante.

Para ilustrar a etapa de mapeamento do documento textual em transações, considere o texto de exemplo sobre *Data Mining*[2] apresentado na Tabela 2.1.

Tabela 2.1: Texto de exemplo.

Data mining is the process of extracting patterns from data. Data mining is becoming an increasingly important tool to transform this data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery. Data mining can be used to uncover patterns in data but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. Data mining cannot discover patterns that may be present in the larger body of data if those patterns are not present in the sample being "mined". Inability to find patterns may become a cause for some disputes between customers and service providers. Therefore data mining is not foolproof but may be useful if sufficiently representative data samples are collected. The discovery of a particular pattern in a particular set of data does not necessarily mean that a pattern is found elsewhere in the larger data from which that sample was drawn. An important part of the process is the verification and validation of patterns on other samples of data.

Para cada documento, o pré-processamento comumente utilizado na Mineração de Textos, como padronização de caixas, remoção de *stopwords* e radicalização das palavras, pode ser aplicado. O resultado do pré-processamento realizado no texto da Tabela 2.1 é apresentado na Tabela 2.2.

Tabela 2.2: Texto da Tabela 2.1 pré-processado.

data mine process extract pattern data. data mine increasingli import tool transform data inform. commonli wide rang profiling practic, such as market, surveil, fraud detect scientif discoveri. data mine uncov pattern data carri sample data. mine process ineffect sample good representat larger bodi data. data mine discov pattern present larger bodi data pattern present sampl mine. inabil find pattern disput custom servic provid. data mine foolproof suffici repres data sampl collect. discoveri pattern set data necessarili pattern found larger data sampl drawn. import part process verif valid pattern sampl data.

Com o documento textual pré-processado, pode-se então mapeá-lo em um conjunto de transações. Nas Tabelas 2.3, 2.4, e 2.5, são apresentadas as transações do texto da Tabela 2.2 mapeadas considerando sentenças, parágrafos e uma janela deslizante de tamanho 10.

Tabela 2.3: Transações obtidas considerando as sentenças do texto da Tabela 2.2.

1.	data mine process extract pattern data
2.	data mine increasingli import tool transform data inform
3.	commonli wide rang profil practic market surveil fraud detect scientif discoveri
4.	data mine uncov pattern data carri sampl data
5.	mine process ineffect sampl good represent larger bodi data
6.	data mine discov pattern present larger bodi data pattern present sampl mine
7.	inabl find pattern disput custom servic provid
8.	data mine foolproof suffici repres data sampl collect
9.	discoveri pattern set data necessarili pattern found larger data sampl drawn
10.	import part process verif valid pattern sampl data

[2]http://en.wikipedia.org/wiki/Data_mining (Acessado em 13 de Maio de 2010)

Tabela 2.4: Transações obtidas considerando os parágrafos da Tabela 2.2.

1. data mine process extract pattern data data mine increasingli import tool transform data inform commonli wide rang profil practic market surveil fraud detect scientif discoveri
2. data mine uncov pattern data carri sampl data mine process ineffect sampl good represent larger bodi data data mine discov pattern present larger bodi data pattern present sampl mine inabl find pattern disput custom servic provid data minefoolproof suffici repres data sampl collect discoveri pattern set data necessarili pattern found larger data sampl drawn import part process verif valid pattern sampl data

Tabela 2.5: Transações obtidas considerando um janela deslizante de tamanho 10 e salto de uma palavra do texto da Tabela 2.2.

1. data
2. data mine
3. data mine process
4. data mine process extract
5. data mine process extract pattern
6. data mine process extract pattern data
7. data mine process extract pattern data data
8. data mine process extract pattern data data mine
9. data mine process extract pattern data data mine increasingli
10. data mine process extract pattern data data mine increasingli import
11. mine process extract pattern data data mine increasingli import tool
12. process extract pattern data data mine increasingli import tool transform
13. extract pattern data data mine increasingli import tool transform data
14. pattern data data mine increasingli import tool transform data inform
15. data data mine increasingli import tool transform data inform commonli
16. data mine increasingli import tool transform data inform commonli wide
17. mine increasingli import tool transform data inform commonli wide rang
18. increasingli import tool transform data inform commonli wide rang profil
19. import tool transform data inform commonli wide rang profil practic
20. . . .
82. data necessarili pattern found larger data sampl drawn import part
83. necessarili pattern found larger data sampl drawn import part process
84. pattern found larger data sampl drawn import part process verif
85. found larger data sampl drawn import part process verif valid
86. larger data sampl drawn import part process verif valid pattern
87. data sampl drawn import part process verif valid pattern sampl
88. sampl drawn import part process verif valid pattern sampl data
89. drawn import part process verif valid pattern sampl data
90. import part process verif valid pattern sampl data
91. part process verif valid pattern sampl data
92. process verif valid pattern sampl data
93. verif valid pattern sampl data
94. valid pattern sampl data
95. pattern sampl data
96. sampl data
97. data

Vale ressaltar que as tarefas de pré-processamento aplicadas nesta etapa são opcionais e vão de acordo com a intenção do usuário. Por exemplo, ao simplificar as palavras de um texto e remover as *stopwords*, pode-se fazer com que *"conjunto de dados"* e *"conjuntos de dados"* sejam contabilizados igualmente, aumentando então a ocorrência de *"conjunt dad"*. Além disso, como o processo de extração de regras de associação não leva em conta a ordem dos itens nas transações, caso seja realizado o pré-processamento no texto, frases como *"mineração dos textos"* e *"textos para mineração"* serão contabilizadas juntas sob a forma *"min text"*. O pré-processamento faz com que frases com variações sintáticas ou

semânticas, mas que em geral representem o mesmo conceito, sejam agregadas no cálculo das frequências.

A forma de modelar o documento em transações pode interferir tanto na frequência das palavras como em suas coocorrências. A frequências das palavras é definida pelo número de ocorrências de uma palavra em uma mesma transação. Por exemplo, a palavra *"data"* ocorre 15 vezes no texto da Tabela 2.2. Ao considerar parágrafos como transações do texto da Tabela 2.2, a palavra *"data"* ocorreu 11 vezes na transação nº2, sendo contabilizada apenas uma vez nesta transação, e ocorreu em todas as transações, tendo sua frequência igual a 2 e suporte de 100%. Já ao considerar as sentenças como transações, a palavra *"data"* apareceu no máximo 3 vezes em uma mesma transação, e ocorreu em 7 das 10 transações, tendo sua frequência igual a 7 e suporte de 70%.

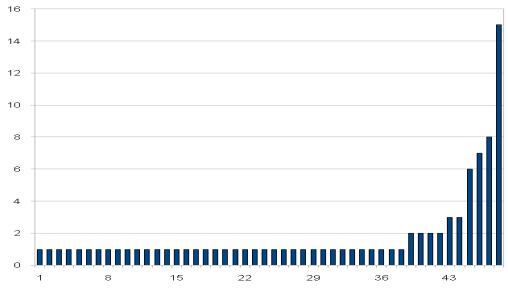
Ao utilizar um mapeamento considerando janelas deslizantes de tamanho pequeno como transações, a distribuição de frequências terá um comportamento semelhante ao da *bag-of-words*. Isso se dá pelo fato que dificilmente uma mesma palavra se repetirá em uma transação modelada por uma janela de tamanho pequeno. Já ao utilizar janelas grandes, uma mesma palavra, principalmente as palavras mais frequentes no documento, tendem a aparecer mais de uma vez em uma mesma transação, fazendo assim com que as diferenças relativas entre as palavras mais frequentes diminuam em relação às demais palavras. Na Figura 2.2 são comparadas as frequências das palavras ao representar o texto por uma *bag-of-words* (a), e por mapear os texto em transações considerando sentenças (b), parágrafos (c), janelas 5 (d), janela 10 (e), janela 20 (f), e janela 30 (g).

As coocorrências das palavras são alteradas conforme muda o tipo de mapeamento. Por exemplo, o mapeamento considerando os parágrafos faz com que as palavras coocorram com mais palavras do que o mapeamento por sentenças. O mesmo ocorre ao utilizar janelas de tamanhos maiores em relação à janelas de tamanhos menores.

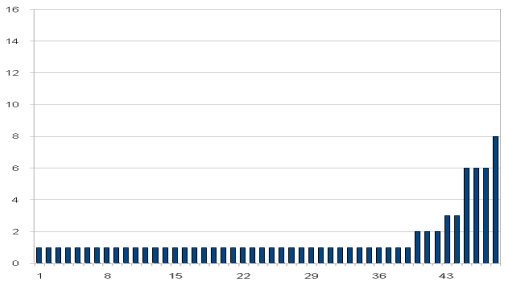
A forma de mapeamento também irá influir no número de transações. Na Tabela 2.6 é apresentado o número de transações obtidas para cada forma de mapeamento analisada para o texto da Tabela 2.2. Pode-se notar uma grande disparidade no número de transações obtidas pelo mapeamento de parágrafos em relação às sentenças, e uma disparidade ainda maior das janelas deslizantes que geraram 800% a mais de transações em relação às sentenças. Portanto, o tipo de mapeamento pode influir no tempo de extração das regras de associação.

2.2 Extraindo Regras de Associação

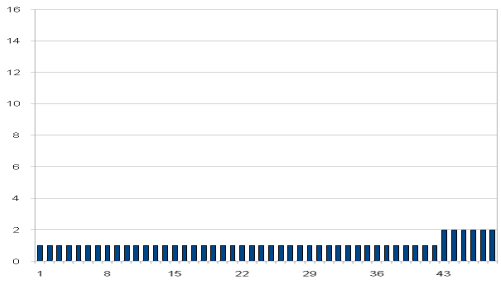
Como o tipo de mapeamento do documento textual em transações pode alterar a distribuição de frequências das palavras, suas coocorrências, o número de transações,



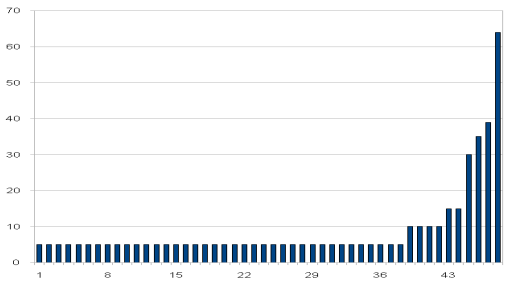
(a) *Bag-of-words*.



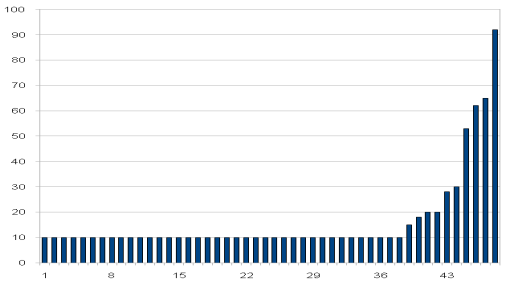
(b) Sentença



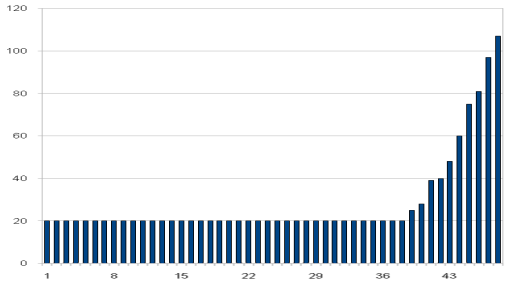
(c) Parágrafo



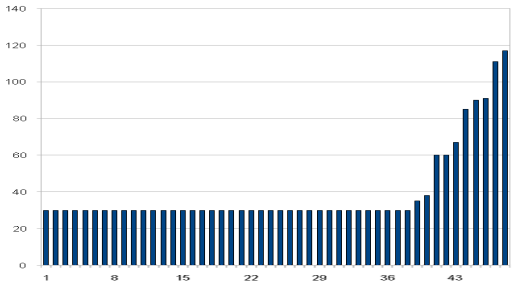
(d) Janela de tamanho 5 e salto 1.



(e) Janela de tamanho 10 e salto 1.



(f) Janela de tamanho 20 e salto 1.



(g) Janela de tamanho 30 e salto 1.

Figura 2.2: Frequência das palavras do texto da Tabela 2.2 na representação *bag-of-words* e nos mapeamentos para transações analisados.

e o valor de suporte mínimo, as relações entre palavras são influenciados pelo tipo de mapeamento.

Para ilustrar o impacto das formas de mapeamento na geração dos *itemsets* frequentes

Tabela 2.6: Número de transações obtidas para o texto da Tabela 2.2 considerando as formas de mapeamento analisadas.

Mapeamento	**Nº de transações**
Sentenças	10
Parágrafos	2
Janela 5	92
Janela 10	97
Janela 20	107
Janela 30	117

e consequentemente nas regras de associação, na Tabela 2.7 são apresentados os *itemsets* extraídos considerando os seguintes mapeamentos e valores de suporte mínimo: $30,0\%$ para o mapeamento de sentença, $100,0\%$ para parágrafo, $10,0\%$ para janela 5, $21,0\%$ para janela 10, $47,0\%$ para janela 20, e $58,0\%$ para janela 30. Esses valores de suporte mínimo foram definidos de forma a gerar o número mais próximo possível de *itemsets* frequentes em todas as formas de mapeamento.

Pode-se notar que alguns *itemsets* importantes para identificar o texto da Tabela 2.1 aparecem em todos os tipos de mapeamento, como *"process"*, *"mine"*, *"pattern"*, *"data"*, *"mine_pattern"*, *"data_mine"*, *"data_pattern"*, e *"data_mine_pattern"* (assinalados em negrito na Tabela 2.7). Alguns *itemsets* possivelmente úteis para representar o texto não aparecem em todos os mapeamentos, como *"mine_sampl"* (apenas na sentença, janela 5, janela 10, e janela 20), *"pattern_sampl"* (só não aparece no mapeamento de parágrafo), *"data_mine_sampl"* (apenas na sentença, janela 10, e janela 20), *"data_mine_process"* (apenas no parágrafo e janela 30), *"mine_pattern_process"* (apenas no parágrafo e janela 30), e *"data_mine_pattern_process"* (apenas no mapeamento de janela 30).

Pode-se notar pelo exemplo da Tabela 2.7 que o suporte mínimo utilizado varia de acordo com o número de transações, o número de itens nas transações e a frequência dos itens. Dado todos esses fatores que fazem que o valor de suporte mínimo varie de acordo com o tipo de transação, tornando difícil a definição do valor de suporte mínimo por parte do usuário, foi proposto em (Rossi, 2011) uma forma para calcular o suporte mínimo automaticamente de cada documento da coleção. Para isso, foram consideradas três premissas:

1. Dado que a frequência total das palavras seja fixa, quanto menor o número de palavras diferentes, maior deve ser o valor de suporte.

2. Dado que o número de transações seja fixo, se a frequência média das palavras aumenta, o suporte deve aumentar.

Tabela 2.7: *Itemsets* frequentes obtidos utilizando as diferentes formas de mapeamentos analisadas para o texto da Tabela 2.2.

Sentença

Itemsets	Suporte
process	30,00%
larger	30,00%
mine	60,00%
pattern	60,00%
sampl	60,00%
data	80,00%
data process	30,00%
larger sampl	30,00%
data larger	30,00%
mine pattern	30,00%
mine sampl	40,00%
data mine	60,00%
pattern sampl	40,00%
data pattern	50,00%
data sampl	60,00%
data larger sampl	30,00%
data mine pattern	30,00%
data mine sampl	40,00%
data pattern sampl	40,00%
-	-

Parágrafo

Itemsets	Suporte
data	100,00%
mine	100,00%
process	100,00%
pattern	100,00%
data mine	100,00%
data process	100,00%
data pattern	100,00%
mine process	100,00%
mine pattern	100,00%
pattern process	100,00%
data mine process	100,00%
data mine pattern	100,00%
data pattern process	100,00%
mine pattern process	100,00%
data mine pattern process	100,00%
-	-
-	-
-	-
-	-
-	-

Janela 5

Itemsets	Suporte
bodi	10,90%
import	10,90%
present	10,90%
discoveri	10,90%
process	16,30%
larger	16,30%
sampl	32,60%
mine	38,00%
pattern	42,40%
data	69,60%
pattern present	9,80%
data larger	12,00%
mine sampl	9,80%
pattern sampl	14,10%
data sampl	21,70%
mine pattern	15,20%
data mine	30,40%
data pattern	30,40%
data pattern sampl	9,80%
data mine pattern	10,90%

Janela 10

Itemsets	Suporte
process	30,90%
larger	28,90%
sampl	54,60%
mine	63,90%
pattern	67,00%
data	94,80%
pattern process	21,60%
data process	30,90%
larger sampl	21,60%
larger pattern	21,60%
data larger	28,90%
mine sampl	30,90%
pattern sampl	42,30%
data sampl	52,60%
mine pattern	44,30%
data mine	61,90%
data pattern	64,90%
data pattern process	21,60%
data larger sampl	21,60%
data larger pattern	21,60%
data mine sampl	28,90%
data pattern sampl	40,20%
data mine pattern	42,30%

Janela 20

Itemsets	Suporte
process	56,10%
sampl	70,10%
mine	75,70%
pattern	90,70%
data	100,00%
pattern process	54,20%
data process	56,10%
mine sampl	47,70%
pattern sampl	69,20%
data sampl	70,10%
mine pattern	69,20%
data mine	75,70%
data pattern	90,70%
data pattern process	54,20%
mine pattern sampl	47,70%
data mine sampl	47,70%
data pattern sampl	69,20%
data mine pattern	69,20%
data mine pattern sampl	47,70%
-	-
-	-
-	-
-	-

Janela 30

Itemsets	Suporte
process	76,90%
sampl	72,60%
mine	77,80%
pattern	94,90%
data	100,00%
mine process	59,80%
pattern process	75,20%
data process	76,90%
pattern sampl	71,80%
data sampl	72,60%
mine pattern	75,20%
data mine	77,80%
data pattern	94,90%
mine pattern process	58,10%
data mine process	59,80%
data pattern process	75,20%
data pattern sampl	71,80%
data mine pattern	75,20%
data mine pattern process	58,10%
-	-
-	-
-	-
-	-

3. Dado que a frequência média das palavras é fixa, quanto maior o número de transações, menor deve ser o suporte.

Com base nessas 3 premissas, a equação para o cálculo automático do suporte mínimo é a seguinte:

$$sup\_aut(d_i) = \frac{(\sum_{\forall a \in A} freq(a)) \ / \ |A|}{|T|} \tag{2.1}$$

na qual A é o conjunto de atributos do documento d_i, $freq(a)$ é a frenquência do atributo $a \in A$, $|A|$ é o número total de atributos e $|T|$ é o número total de transações do documento d_i.

O numerador da Equação 2.1 atende às duas primeiras premissas, ou seja, se o número de palavras diferentes aumentar, o suporte irá diminuir, uma vez que há uma ponderação

pelo número de palavras diferentes ($|A|$). Se a frequência média das palavras aumenta, o valor de suporte também aumenta, uma vez que o numerador da Equação 2.1 refere-se a frequência média. Já o numerador desta equação atende à terceira premissa, ou seja, quanto maior o número de transações, menor será o valor de suporte, uma vez que há uma ponderação pelo número de transações ($|\mathrm{T}|$).

O suporte mínimo automático isenta o usuário de conhecer as características do documento ou da coleção de documentos. Além disso, evita-se que o usuário defina um valor de suporte mínimo baixo de forma a gerar uma grande quantidade de regras, já que a fórmula para o cálculo do suporte mínimo leva em consideração a frequência média das palavras nas transações.

Embora as medidas de suporte e confiança sejam as mais comuns na extração de regras de associação, em muitos casos as regras obtidas utilizando somente essas medidas podem não apresentar relacionamentos interessantes, mesmo com altos valores de confiança. Isto acontece principalmente quando o valor de suporte do antecedente é baixo e o valor de suporte do consequente é alto. Neste caso, os itens que mais ocorrerem nas transações compõem a maioria das regras. Por exemplo, suponha um texto mapeado em 10 transações, e que foi extraída a regra $stemming \rightarrow clustering$, cujo suporte de $stemming$ é $10,0\%$, de $clusterging$ é $90,0\%$ e de $clustering \cup stemming$ é 10%. Neste caso, a confiança da regra $stemming \rightarrow clustering$ seria de $100,0\%$, mesmo que a palavra $stemming$ tenha aparecido uma única vez e que nesta vez tenha coocorrido com a palavra $clustering$.

Ainda há casos em que os *itemsets* das regras podem ser independentes, ou seja, a presença de um *itemset* não aumenta nem diminui as chances de ocorrência de um outro *itemset*. Por exemplo, suponha novamente um texto mapeado em 10 transações, e que foi extraída a regra $information \rightarrow categorization$, cujo suporte de $information$ é $50,0\%$, de $categorization$ é $80,0\%$, e de $categorization \cup information$ é 40%. Neste caso, a confiança da regra $information \rightarrow categorization$ seria de $80,0\%$, que é o mesmo valor do suporte do item $categorizatioin$, ou seja, a probabilidade de ocorrer $categorization$ dado que $information$ ocorreu é a mesma de ocorrer somente $categorization$.

Além das medidas de suporte e confiança, outras medidas, conhecidas como medidas de interesse, podem ser utilizadas para identificar regras não triviais, relevantes, e úteis ao usuário, pois capturam diferentes relações entre itens que não são capturadas pelas medidas suporte e confiança. Como a representação *bag-of-related-words* não visa utilizar conhecimento de domínio, são utilizadas medidas de interesse "objetivas", ou seja, que consideram somente o conjunto de dados para calcular o valor de uma regra. O objetivo de utilizar essas medidas é obter atributos mais significativos e diminuir ainda mais a dimensionalidade da representação obtida.

De acordo com suas características, as medidas de interesse objetivas podem ser con-

flitantes quanto ao ranqueamento das regras, e consequentemente na poda. Algumas medidas de interesse objetivas podem ser melhores para alguns domínios de aplicação do que outras. Entretanto, pode haver situações em que as medidas podem ser consistentes umas com as outras. Para ilustrar essas situações, na Tabela 2.8 são apresentados os *rankings* de 25 atributos extraídos de um texto sobre medidas de validação de agrupamento de dados (Wu et al., 2009) gerados pelas medidas de interesse objetivas `Suporte`, `Confiança`, `Lift`, `Yules'Q`, `Coeficiente de Correlação Linear`, `Informação Mútua`, `Gini Index`, `Kappa`, e `J-Measure` (Guillet e Hamilton, 2007; Geng e Hamilton, 2006; Blanchard et al., 2005; Tan et al., 2002).

Pode-se notar que as medidas derivadas da estatística e que medem a dependência entre os itens da regra, como Lift, Yule's Q e Correlação, obtiveram a maior pontuação para o *itemset* "*contig matrix*". Já as medidas baseadas em teoria da informação, como Informação Mútua e J-Measure, obtiveram a maior pontuação para o *itemset* "*data set*".

Ao comparar o *ranking* obtido pelas medidas Informação Mútua e J-Measure na Tabela 2.8, ambas originárias da teoria da informação, pode-se notar uma grande semelhança entre os itens que compõem o *ranking*. Ao todo são 23 itens em comum dos 25 itens que compões o *ranking*. Porém, alguns elementos, como *effect_mean_uniform*, aparecem em posições diferentes do *ranking*. Além disso, atributos importantes como *measur_norm* e *cluster_evalu* só apareceram no *ranking* da medida Informação Mútua, enquanto que *cluster_measur_valid* e *extern_measur_valid* só apareceram no *ranking* da medida J-Measure.

De maneira semelhante, ao analisar as medidas Yule's Q e Correlação na Tabela 2.8, alguns atributos como *defect_measur* aparecem em posições distantes no *ranking* de cada medida. Atributos importantes como *extern_measur_valid*, e *measur_puriti* só apareceram no *ranking* da medida Yule's Q, enquanto que atributos como *cluster_valid* e *measur_normal* só apareceram no *ranking* da medida Correlação.

Já ao comparar medidas como a Confiança, que analisa a força da implicação da regra, e a medida Lift, que analisa as dependências dos itens da regra, pode-se notar uma grande diferença dos atributos e das posições do *ranking*.

2.3 Gerando os Atributos e Construindo a Matriz Atributo-Valor

Para gerar os atributos da representação *bag-of-related-words* são utilizados os *itemsets* das regras de associação extraídas. Por exemplo, se fosse extraída a regra $data \rightarrow mine$, seria gerado o atributo "*data_mine*".

Para evitar situações em que regras diferentes compostas pelos mesmos *itemsets* como $mine \rightarrow data$ e $data \rightarrow mine$ gerem atributos diferentes, pode-se ordenar os itens em ordem lexicográfica ou na ordem em que eles aparecem no texto. Na abordagem proposta,

não é feito o enriquecimento da abordagem *bag-of-words*, mas são utilizados os 1-*itemsets* frequentes como atributos compostos por palavras simples.

Uma vez gerados os atributos dos documentos utilizando as regras extraídas, é possível gerar a representação no modelo espaço vetorial. Os pesos dos atributos na matriz podem ser os mesmos utilizados comumente na Mineração de Textos ou pode-se utilizar o suporte da regra que foi utilizada para gerar o atributo.

No próximo capítulo é apresentada a ferramenta FEATuRE, que realiza todos os passos apresentados neste capítulo para a geração da representação *bag-of-related-words*.

Tabela 2.8: *Ranking* dos atributos segundo as medidas de interesse objetivas utilizadas neste trabalho.

Suporte

Rank	**Atributos**	*Score*
1º	measur valid	0,073
2º	data set	0,055
3º	cluster measur	0,053
4º	measur normal	0,050
5º	cluster valid	0,046
6º	cluster mean	0,039
7º	cluster result	0,036
8º	measur properti	0,027
9º	class cluster	0,025
10º	cluster measur valid	0,024
11º	data sampl	0,020
12º	extern measur	0,020
13º	mean measur	0,020
14º	effect uniform	0,019
15º	extern valid	0,019
16º	conting matrix	0,018
17º	result valid	0,018
18º	cluster evalu	0,017
19º	class data	0,017
20º	entropi measur	0,016
21º	cluster number	0,016
22º	bound upper	0,015
23º	effect mean	0,015
24º	evalu measur	0,015
25º	class distribut	0,015

Confiança

Rank	**Atributos**	*Score*
1º	conting matrix	0,766
2º	effect uniform	0,663
3º	effect mean uniform	0,600
4º	bound upper	0,562
5º	data set	0,552
6º	cluster measur valid	0,336
7º	inform mutual	0,331
8º	rand statist	0,271
9º	measur valid	0,237
10º	data sampl set	0,226
11º	cluster valid	0,207
12º	cluster mean measur	0,203
13º	data sampl	0,199
14º	effect mean	0,198
15º	sampl set	0,189
16º	extern measur valid	0,184
17º	mean uniform	0,179
18º	class distribut	0,179
19º	cluster mean	0,175
20º	class data	0,172
21º	cluster measur	0,172
22º	extern valid	0,171
23º	result valid	0,166
24º	measur normal	0,162
25º	class set	0,161

Lift

Rank	**Atributos**	*Score*
1º	conting matrix	37,267
2º	bound upper	30,864
3º	effect uniform	23,399
4º	inform mutual	21,666
5º	rand statist	20,242
6º	data sampl set	8,783
7º	data imbalanc	8,125
8º	effect mean uniform	8,097
9º	data simul	8,000
10º	data sampl	7,692
11º	data set	7,432
12º	sampl set	7,276
13º	effect mean	6,868
14º	mean uniform	6,189
15º	class size	6,172
16º	extern valid	5,187
17º	class distribut	4,306
18º	defect measur	2,961
19º	data distribut	2,558
20º	cluster evalu	2,518
21º	result valid	2,457
22º	cluster number	2,452
23º	measur section	2,443
24º	cluster result	2,424
25º	cluster perform	2,311

Yule's Q

Rank	**Atributos**	*Score*
1º	conting matrix	0,998
2º	bound upper	0,995
3º	inform mutual	0,991
4º	effect uniform	0,990
5º	rand statist	0,982
6º	data set	0,965
7º	data imbalanc	0,956
8º	data simul	0,952
9º	data sampl	0,947
10º	effect mean uniform	0,919
11º	defect measur	0,917
12º	data sampl set	0,913
13º	sampl set	0,893
14º	effect mean	0,886
15º	class size	0,860
16º	mean uniform	0,859
17º	extern valid	0,856
18º	class distribut	0,757
19º	measur section	0,750
20º	measur valid	0,686
21º	extern measur valid	0,670
22º	measur properti	0,669
23º	cluster evalu	0,654
24º	cluster result	0,650
25º	measur puriti	0,646

Correlação

Rank	**Atributos**	*Score*
1º	conting matrix	0,814
2º	bound upper	0,673
3º	effect uniform	0,657
4º	data set	0,606
5º	inform mutual	0,520
6º	rand statist	0,438
7º	data sampl	0,364
8º	data sampl set	0,329
9º	data imbalanc	0,302
10º	sampl set	0,289
11º	effect mean	0,289
12º	data simul	0,287
13º	effect mean uniform	0,287
14º	extern valid	0,273
15º	measur valid	0,268
16º	mean uniform	0,260
17º	class size	0,250
18º	class distribut	0,208
19º	cluster result	0,204
20º	cluster mean	0,191
21º	measur properti	0,162
22º	cluster valid	0,160
23º	cluster evalu	0,143
24º	measur normal	0,139
25º	defect measur	0,137

Informação Mútua

Rank	**Atributos**	*Score*
1º	data set	0,148
2º	conting matrix	0,100
3º	effect uniform	0,083
4º	bound upper	0,073
5º	inform mutual	0,055
6º	data sampl	0,049
7º	measur valid	0,047
8º	rand statist	0,039
9º	data imbalanc	0,033
10º	extern valid	0,032
11º	data sampl set	0,031
12º	effect mean	0,030
13º	data simul	0,030
14º	sampl set	0,029
15º	effect mean uniform	0,028
16º	mean uniform	0,025
17º	cluster result	0,024
18º	class size	0,023
19º	cluster mean	0,022
20º	class distribut	0,019
21º	measur properti	0,017
22º	cluster valid	0,016
23º	measur normal	0,013
24º	defect measur	0,012
25º	cluster evalu	0,012

Gini Index

Rank	**Atributos**	*Score*
1º	data set	0,066
2º	measur valid	0,030
3º	conting matrix	0,029
4º	effect uniform	0,024
5º	data sampl	0,023
6º	bound upper	0,023
7º	inform mutual	0,020
8º	effect mean uniform	0,019
9º	data sampl set	0,019
10º	data imbalanc	0,016
11º	data simul	0,014
12º	extern valid	0,014
13º	cluster result	0,014
14º	rand statist	0,014
15º	cluster measur valid	0,013
16º	cluster mean	0,012
17º	effect mean	0,012
18º	extern measur valid	0,011
19º	sampl set	0,011
20º	measur properti	0,011
21º	mean uniform	0,009
22º	class size	0,009
23º	cluster valid	0,008
24º	measur normal	0,008
25º	defect measur	0,008

Kappa

Rank	**Atributos**	*Score*
1º	measur valid	0,427
2º	measur normal	0,384
3º	cluster measur	0,361
4º	measur properti	0,351
5º	extern measur	0,339
6º	entropi measur	0,333
7º	evalu measur	0,330
8º	mean measur	0,329
9º	measur select	0,328
10º	cluster valid	0,328
11º	consist measur	0,327
12º	measur section	0,327
13º	measur puriti	0,326
14º	defect measur	0,324
15º	equival measur	0,323
16º	measur result	0,321
17º	inform measur	0,319
18º	cluster mean	0,310
19º	cluster result	0,303
20º	class cluster	0,283
21º	cluster evalu	0,262
22º	cluster data	0,261
23º	cluster number	0,259
24º	data set	0,254
25º	cluster set	0,254

J-Measure

Rank	**Atributos**	*Score*
1º	data set	0.124
2º	conting matrix	0.085
3º	effect uniform	0.072
4º	bound upper	0.066
5º	inform mutual	0.051
6º	data sampl	0.047
7º	effect mean uniform	0.046
8º	measur valid	0.041
9º	rand statist	0.037
10º	data imbalanc	0.032
11º	extern valid	0.030
12º	data simul	0.029
13º	data sampl set	0.029
14º	effect mean	0.028
15º	sampl set	0.028
16º	cluster measur valid	0.024
17º	mean uniform	0.024
18º	cluster result	0.022
19º	class size	0.022
20º	cluster mean	0.020
21º	extern measur valid	0.019
22º	class distribut	0.017
23º	measur properti	0.016
24º	cluster valid	0.014
25º	defect measur	0.012

Capítulo 3

Ferramenta computacional desenvolvida para a geração da representação *bag-of-related-words*

Para apoiar todos os passos da geração da representação *bag-of-related-words*, foi desenvolvida a ferramenta FEATuRE (Features gEnerator based on AssociaTion RulEs). A ferramenta foi desenvolvida na linguagem de programação Java e implementa todos os passos apresentados no Capítulo 2. A FEATuRE está disponível em `http://sites.labic.icmc.usp.br/feature/distribuicao/`.

Na Figura 3.1 é apresentada a tela principal na ferramenta FEATuRE. No painel `Step by Step` encontram-se as funcionalidades referentes aos passos definidos no Capítulo 2. No painel `Utilities` encontram os utilitário para auxiliar o usuário na geração, análise e uso da representação *bag-of-related-words*, que serão detalhados na Seção 3.2.

Nas próximas seções serão apresentadas as telas para a geração da representação *bag-of-related-words*, dos utilitários da ferramenta FEATuRE, e detalhes da implementação.

3.1 Passo a passo da representação *bag-of-related-words*

Nesta seção serão apresentadas as funcionalidades para a geração da representação *bag-of-related-words*.

3.1.1 Mapeamento de um documento textual em transações

Na Figura 3.2 é apresentada a tela para realizar o mapeamento dos documentos textuais em transações. É necessário informar o diretório contendo os documentos textuais os quais se deseja realizar a estruturação no modelo espaço vetorial na caixa `Input`, e o diretório de saída na caixa `Output`, o qual serão gravados os arquivos de transações correspondentes aos arquivos textos. Portando, se existir um arquivo nomeado "`texto_1.txt`"

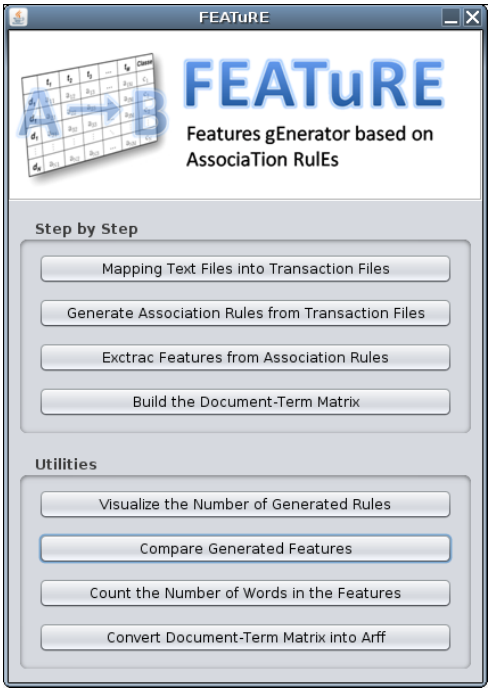


Figura 3.1: FEATuRE - Tela inicial.

no diretório `Input`, será gravado um arquivo também nomeado como "`texto_1.txt`" no diretório `Output`.

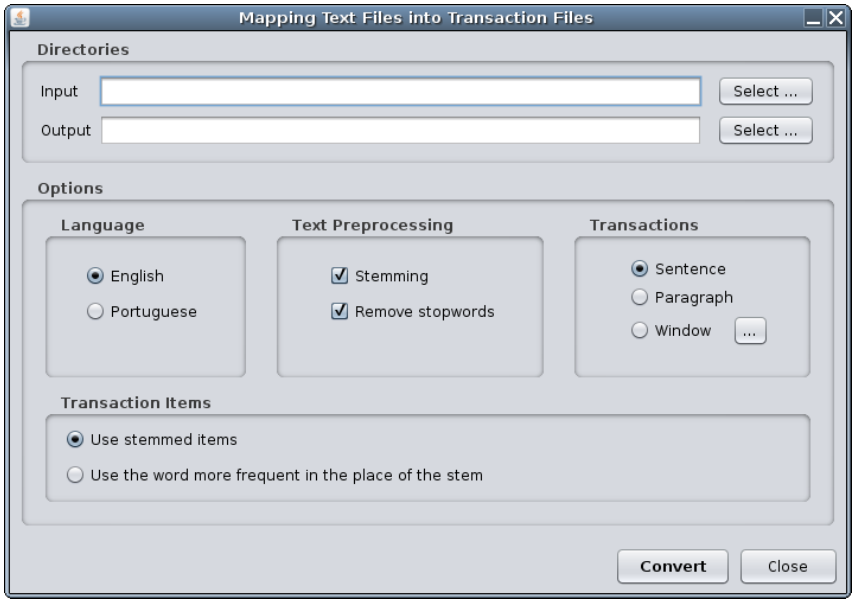


Figura 3.2: FEATuRE - Tela para conversão dos documentos textuais em arquivos de transações.

A ferramenta suporta documentos textuais escritos em duas línguas: inglês e português. A linguagem é especificada para que se possa realizar o processo de radicalização das palavras.

Na fase de pré-processamento, o usuário pode optar por realizar a remoção de *stopwords* e a radicalização de palavras. A radicalização faz com que palavras como `agrupar`, `agrupamento` e `agrupado` sejam representadas por uma única forma, neste caso o radical `agrup`. A padronização de caixas é definida como padrão, ou seja, todas as letras são transformadas para minúsculas. Caso o usuário opte por radicalizar as palavras, no painel `Transaction Items`, o usuário pode selecionar a opção `Use the more frequent word in the place of the stem` para substituir as palavras radicalizadas pela palavras mais frequente que originou o radical.

No painel `Transactions` é definido a maneira como o documento será mapeado em transações. São definidas 3 formas: sentenças, parágrafos, e janela deslizante. Quando for selecionada a opção janela deslizante (`Window`), é possível definir o tamanho da janela e o salto, clicando sobre o botão com o rótulo "`...`".

Na Tabela 3.1 é ilustrado o arquivo de entrada e a saída gerada pela ferramenta FEATuRE. Como entrada, é dado um arquivo em formato texto. Neste exemplo, foi considerado um texto sobre Mineração de Textos retirado da Wikipedia[1]. Como saída, são geradas as transações considerando o mapeamento de janela deslizante de tamanho 5, em que cada linha do arquivo representa uma transação.

3.1.2 Extraindo regras de associações das transações mapeadas

Na Figura 3.2 é apresentada a tela para extração de regras de associação dos arquivos de transações. É necessário informar o diretório contendo os arquivos de transações na caixa `Input`, e o diretório de saída na caixa `Output`, o qual serão gravados os arquivos contendo as regras extraídas de cada arquivo de transações. Portando, se existir um arquivo de transações nomeado "`texto_1.txt`" no diretório `Input`, será gravado um arquivo de regras também nomeado como "`texto_1.txt`" no diretório `Output`.

Nesta etapa é definido o valor do limiar de suporte, podendo este ser definido manualmente ou pela equação de suporte automático apresentado na Equação 2.1. Também é possível definir nesta etapa um valor de um multiplicador (α) para aumentar ou diminuir o valor de limiar gerado pelo suporte automático. Para extrair as regras de associação foi utilizada a implementação do algoritmo APRIORI desenvolvida por (Borgelt, 2004).

Na Tabela 3.2 é ilustrado o arquivo de entrada e saída para esta etapa da ferramenta FEATuRE. Como entrada, é dado um arquivo de transações e como saída são geradas as regras de associação, onde cada linha do arquivo representa uma regra e as valores entre parênteses representam o suporte e a confiança respectivamente.

[1] `http://en.wikipedia.org/wiki/Text_mining`, 27 de julho de 2011.

Tabela 3.1: Exemplo de mapeamento de um documento textual gerado pelo ferramenta FEATuRE considerando um texto sobre Mineração de Textos como entrada e uma janela deslizante de tamanho 5.

Entrada	Saída
Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high- quality information from text. High-quality in- formation is typically derived through the divising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and in- terpretation of the output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities). . . . Text mining is an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics. . . . Text mining is being used by large media companies, such as the Tribune Company, to disambiguate information and to provide readers with greater search experiences, which in turn increases site "stickiness"and revenue. . . . Until recently, websites most often used text-based searches, which only found documents containing specific user-defined words or phrases. Now, through use of a semantic web, text mining can find content based on meaning and context (rather than just by a specific word). Additionally, text mining software can be used to build large dossiers of information about specific people and events. For example, large datasets based on data extracted from news reports can be built to facilitate social networks analysis or counter-intelligence. In effect, the text mining software may act in a capacity similar to an intelligence analyst or research librarian, albeit with a more limited scope of analysis. Text mining is also used in some email spam filters as a way of determining the characteristics of messages that are likely to be advertisements or other unwanted material.	text text mine text mine altern text mine altern refer text mine altern refer text mine altern refer text data altern refer text data mine refer text data mine roughli text data mine roughli equival data mine roughli equival text mine roughli equival text analyt . . . text mine interdisciplinari field draw mine interdisciplinari field draw inform interdisciplinari field draw inform retriev field draw inform retriev data draw inform retriev data mine inform retriev data mine machin retriev data mine machin learn . . . text mine larg media compani mine larg media compani tribun larg media compani tribun compani media compani tribun compani disambigu compani tribun compani disambigu inform tribun compani disambigu inform provid . . . recent websit text base search websit text base search found text base search found document base search found document specif search found document specif user . . . email spam filter determin characterist spam filter determin characterist messag filter determin characterist messag advertis determin characterist messag advertis unwant characterist messag advertis unwant materi messag advertis unwant materi advertis unwant materi unwant materi materi

Figura 3.3: FEATuRE - Tela para conversão dos arquivos de transações em arquivos de regras de associação.

3.1.3 Utilizando os itens das regras para compor os atributos

Na Figura 3.4 é apresentada a tela para a geração de atributos à partir das regras de associação. No painel `Strategy for feature generation` são definidas as medidas objetivas que são utilizadas para podar os atributos, ou os tipos de *itemsets* frequentes que serão utilizados como atributos: *itemsets* e *maximal itemsets*. Também neste painel pode ser definido um novo valor de suporte mínimo e o limiar da medida objetiva seleci-

Tabela 3.2: Exemplo de mapeamento de um documento textual gerado pelo ferramenta FEATuRE considerando as sentenças como transações.

Entrada	Saída
text	biomed <- (3.2, 3.2)
text mine	pattern <- (3.2, 3.2)
text mine altern	learn <- (3.5, 3.5)
text mine altern refer	entiti <- (3.5, 3.5)
text mine altern refer text	larg <- (3.5, 3.5)
mine altern refer text data	analyt <- (3.5, 3.5)
altern refer text data mine	...
refer text data mine roughli	data <- (5.8, 5.8)
text data mine roughli equival	softwar <- (5.8, 5.8)
data mine roughli equival text	analysi <- (6.9, 6.9)
mine roughli equival text analyt	research <- (6.9, 6.9)
roughli equival text analyt refer	inform <- (10.1, 10.1)
equival text analyt refer process	mine <- (33.2, 33.2)
...	text <- (37.6, 37.6)
text mine interdisciplinari field draw	...
mine interdisciplinari field draw inform	learn <- statist (1.6, 70.0)
interdisciplinari field draw inform retriev	statist <- learn (1.6, 46.7)
field draw inform retriev data	mine <- semant (1.6, 70.0)
draw inform retriev data mine	semant <- mine (1.6, 4.9)
inform retriev data mine machin	text <- semant (1.8, 80.0)
retriev data mine machin learn	semant <- text (1.8, 4.9)
data mine machin learn statist	...
mine machin learn statist comput	high <- qualiti (2.8, 85.7)
...	qualiti <- high (2.8, 63.2)
text mine larg media compani	inform <- qualiti (1.8, 57.1)
mine larg media compani tribun	qualiti <- inform (1.8, 18.2)
larg media compani tribun compani	text <- qualiti (2.1, 64.3)
media compani tribun compani disambigu	qualiti <- text (2.1, 5.5)
compani tribun compani disambigu inform	mine <- biomed (2.1, 64.3)
tribun compani disambigu inform provid	biomed <- mine (2.1, 6.3)
compani disambigu inform provid reader	text <- biomed (1.8, 57.1)
disambigu inform provid reader greater	...
inform provid reader greater search	mine <- data (3.0, 52.0)
...	data <- mine (3.0, 9.0)
recent websit text base search	mine <- softwar (3.5, 60.0)
websit text base search found	softwar <- mine (3.5, 10.4)
text base search found document	text <- softwar (2.5, 44.0)
base search found document specif	softwar <- text (2.5, 6.7)
search found document specif user	mine <- analysi (2.5, 36.7)
found document specif user defin	analysi <- mine (2.5, 7.6)
document specif user defin word	text <- analysi (3.0, 43.3)
...	analysi <- text (3.0, 8.0)
email spam filter determin characterist	text <- mine (25.8, 77.8)
spam filter determin characterist messag	mine <- text (25.8, 68.7)
filter determin characterist messag advertis	...
determin characterist messag advertis unwant	text <- biomed mine (1.6, 77.8)
characterist messag advertis unwant materi	mine <- biomed text (1.6, 87.5)
messag advertis unwant materi	biomed <- mine text (1.6, 6.3)
advertis unwant materi	text <- softwar mine (2.5, 73.3)
unwant materi	mine <- softwar text (2.5, 100.0)
materi	softwar <- mine text (2.5, 9.8)

onada. Vale ressaltar que como os possíveis valores da medida *Lift* variam entre 0 e ∞, na ferramenta foi implementada a padronização proposta em McNicholas et al. (2008), fazendo com que os valores dessa medida variem enter 0 e 1.

No painel `Options`, da Figura 3.4, o usuário pode repetir os atributos no arquivo resultante do processo de acordo com o valor de suporte da regra que originou o atributo por meio da opção `Repeat features according to the support`. Os valores de suporte são arredondados sempre para cima. Por exemplo, se a regra "$mine \Rightarrow text$" tem suporte de $24,6\%$, o atributo "*mine_text*" aparecerá 25 vezes no arquivo de atributos.

Também é possível optar por um limiar de medida objetiva definido automaticamente. Para isto, basta selecionar a opção `Obtain objective measure threshold automatically`. A ferramenta FEATuRE calcula a média dos valores da medida objetiva para todas as regras de um determinado documento e utiliza esta média como limiar.

Pode-se optar também por utilizar apenas as X regras mais bem ranqueadas segundo alguma medida objetiva disponível na ferramenta. Para isto basta selecionar a opção `Use`

de X best ranked rules e definir o valor de X.

É necessário informar o diretório contendo os arquivos de regras de associação os quais se desejam gerar os atributos na caixa Input, e o diretório de saída na caixa Output, o qual são gravados os arquivos contendo os atributos gerado à partir de cada arquivo de regras de associação. Portando, se existir um arquivo de regras nomeado "texto_1.txt" no diretório Input, será gravado um arquivo de atributos também nomeado como "texto_1.txt" no diretório Output.

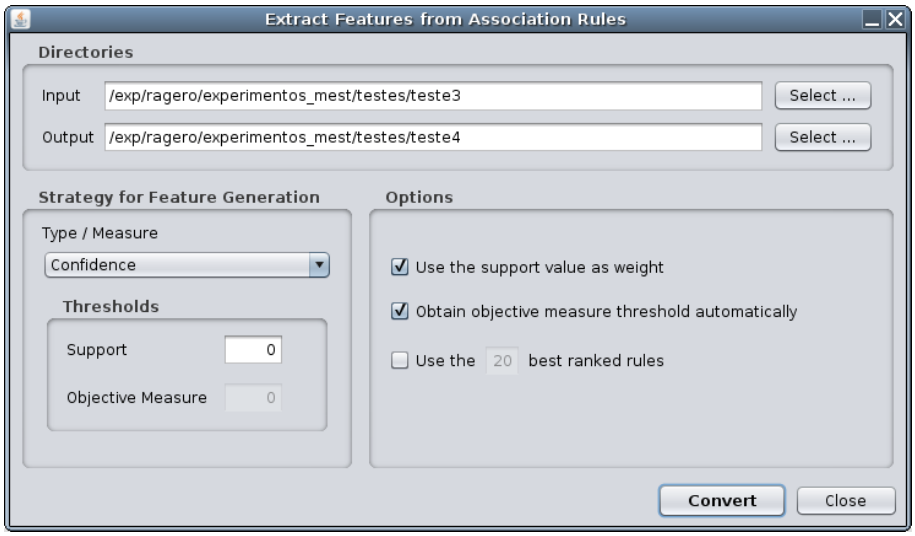


Figura 3.4: FEATuRE - Tela para conversão dos arquivos de regras de associação em arquivos de atributos.

Na Tabela 3.3 é ilustrado o arquivo de entrada e saída desta etapa da ferramenta. Como entrada, é dado um arquivo de regras de associação e como saída são gerados atributos, onde cada linha do arquivo representa um atributo. Os atributos da Tabela 3.3 foram gerados utilizando a medida Informação Mútua com um limiar de $0,06$.

3.1.4 Utilizando os atributos para construir uma representação no modelo espaço vetorial

Na Figura 3.5 é apresentada a tela para gerar uma representação no espaço vetorial utilizando os arquivos de atributos gerados no passo anterior. Caso os arquivos de atributos estejam separados em pastas, é possível utilizar estas como sendo as classes dos documentos por meio da opção Consider the folder as file class.

É necessário informar o diretório contendo os arquivos de atributos os quais se deseja representar em um espaço vetorial na caixa Input, e o diretório de saída na caixa Output, o qual são gravados os arquivos "discover.names" e "discover.data".

Tabela 3.3: Exemplo de mapeamento de um documento textual gerado pelo ferramenta FEATuRE considerando as sentenças como transações.

Entrada	Saída
biomed <- (3.2, 3.2)	compani
pattern <- (3.2, 3.2)	advanc
learn <- (3.5, 3.5)	relat
entiti <- (3.5, 3.5)	model
larg <- (3.5, 3.5)	includ
analyt <- (3.5, 3.5)	cluster
...	system
data <- (5.8, 5.8)	field
softwar <- (5.8, 5.8)	combin
analysi <- (6.9, 6.9)	statist
research <- (6.9, 6.9)	academ
inform <- (10.1, 10.1)	network
mine <- (33.2, 33.2)	scienc
text <- (37.6, 37.6)	social
...	intellig
learn <- statist (1.6, 70.0)	...
statist <- learn (1.6, 46.7)	structur
mine <- semant (1.6, 70.0)	internet
semant <- mine (1.6, 4.9)	recent
text <- semant (1.8, 80.0)	content
semant <- text (1.8, 4.9)	custom
...	qualiti
high <- qualiti (2.8, 85.7)	biomed
qualiti <- high (2.8, 63.2)	pattern
inform <- qualiti (1.8, 57.1)	learn
qualiti <- inform (1.8, 18.2)	refer
text <- qualiti (2.1, 64.3)	mean
qualiti <- text (2.1, 5.5)	entiti
mine <- biomed (2.1, 64.3)	larg
biomed <- mine (2.1, 6.3)	analyt
text <- biomed (1.8, 57.1)	base
...	...
mine <- data (3.0, 52.0)	univers
data <- mine (3.0, 9.0)	similar
mine <- softwar (3.5, 60.0)	involv
softwar <- mine (3.5, 10.4)	area
text <- softwar (2.5, 44.0)	linguist
softwar <- text (2.5, 6.7)	high
mine <- analysi (2.5, 36.7)	deriv
analysi <- mine (2.5, 7.6)	specif
text <- analysi (3.0, 43.3)	data
analysi <- text (3.0, 8.0)	softwar
text <- mine (25.8, 77.8)	analysi
mine <- text (25.8, 68.7)	research
...	inform
text <- biomed mine (1.6, 77.8)	mine
mine <- biomed text (1.6, 87.5)	text
biomed <- mine text (1.6, 6.3)	entiti_relat
text <- softwar mine (2.5, 73.3)	learn_statist
mine <- softwar text (2.5, 100.0)	high_qualiti
softwar <- mine text (2.5, 9.8)	mine_text

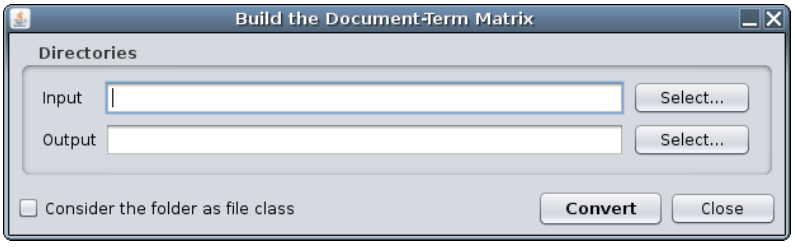


Figura 3.5: FEATuRE - Tela para conversão dos arquivos de atributos em uma matriz atributos valor.

O arquivo "`discover.names`" contém a lista dos atributos que estão representados no modelo espaço vetorial. Cada linha contém o atributo escrito entre aspas (" "), seguido pelo sinal de dois pontos (:), depois pelo tipo de valor usado para definir o peso do atributo, seguido pelo sinal de ponto final (.). A primeira linha desse arquivo corresponde ao atributo referente ao identificador do documento textual e as demais linhas representam os atributos gerados. Caso os documentos possuam classe, a primeira linha conterá a *string*

"`att_class.`" e a última linha conterá os possíveis valores do atributo classe. A diferença entre o arquivo "`discover.names`" sem e com a informação de classe é apresentada na Tabela 3.4.

Tabela 3.4: Exemplo de mapeamento de um documento textual gerado pelo ferramenta FEATuRE considerando as sentenças como transações.

Sem Classe	**Com Classe**
filename:string:ignore.	att_class.
"high_inform_qualiti":integer.	filename:string:ignore.
"deriv":integer.	"high_inform_qualiti":integer.
"intellig":integer.	"deriv":integer.
"deriv_pattern":integer.	"intellig":integer.
"univers":integer.	...
...	att_class:nominal("Classe1","Classe2","Classe3").

No arquivo "`discover.data`", cala linha corresponde aos valores atribuídos a cada documento da coleção. Cada valor é separado pelo sinal de vírgula (,). O primeiro valor corresponde ao nome do arquivo ao qual foram gerados os atributos. Os demais valores representam os pesos dados a cada atributo em um determinado documento da coleção. Caso os documentos da coleção possuam informação de classe, o último valor da linha corresponderá à classe.

Na Tabela 3.5 são ilustrados os arquivos de entrada e saída desta etapa da ferramenta FEATuRE. Como entrada, são dados arquivos de atributos e como saída são gerados os arquivos "`discover.names`" e "`discover.data`".

3.2 Utilitários da ferramenta FEATuRE

Ainda há uma série de outros utilitários implementados na ferramenta FEATuRE, como i) visualização do número de regras geradas para cada documento da coleção, ii) comparação dos atributos de diferentes representações, iii) contagem de quantos atributos são compostos por n palavras, e iv) conversão da uma matriz atributo-valor gerada pela ferramenta FEATuRE para o formato *arff*, utilizado na ferramenta Weka (Witten e Frank, 2005). A seguir serão apresentadas as telas dos utilitários do sistema.

3.2.1 Visualizando o número de regras geradas

Durante a geração da representação *bag-of-related-words* é importante verificar o número de regras que estão sendo geradas para se ter uma idéia da dimensionalidade da representação que será gerada. Na Figura 3.6 é apresentada a tela para visualizar o número de regras geradas para os documentos textuais de uma coleção. Para isso é necessário informar o diretório contendo os arquivos de regras na caixa `Directory` e clicar no botão

Tabela 3.5: Exemplo de mapeamento de um documento textual gerado pelo ferramenta FEATuRE considerando as sentenças como transações.

Entrada	**Saída**
arquivo_1	**discover.names**
text_mine	att_class.
data_mine	filename:string:ignore.
machine_learn	”text_mine”:integer.
inform_retriev	”data_mine”:integer.
arquivo_2	”machine_learn”:integer.
text_mining	”inform_retriev”:integer.
cluster	”cluster”:integer.
inform_retriev	”topic_hierarch”:integer.
...	”classif”:integer.
arquivo_N	att_class:nominal(”Classe_X”,”Classe_Y”,...,”Classe_Z”).
text_mine	**discover.data**
topic_hierarch	“arquivo_1”,1,1,1,1,0,0,0,Classe_X
classif	“arquivo_2”,1,0,0,1,1,0,0,Classe_Y
cluster	...
machine_learn	“arquivo_N”,1,0,1,0,1,1,1,Classe_Z

`Visualize`. Após a análise é informado o número de regras em cada arquivo, o total e o número médio de regras de todos os arquivos do diretório.

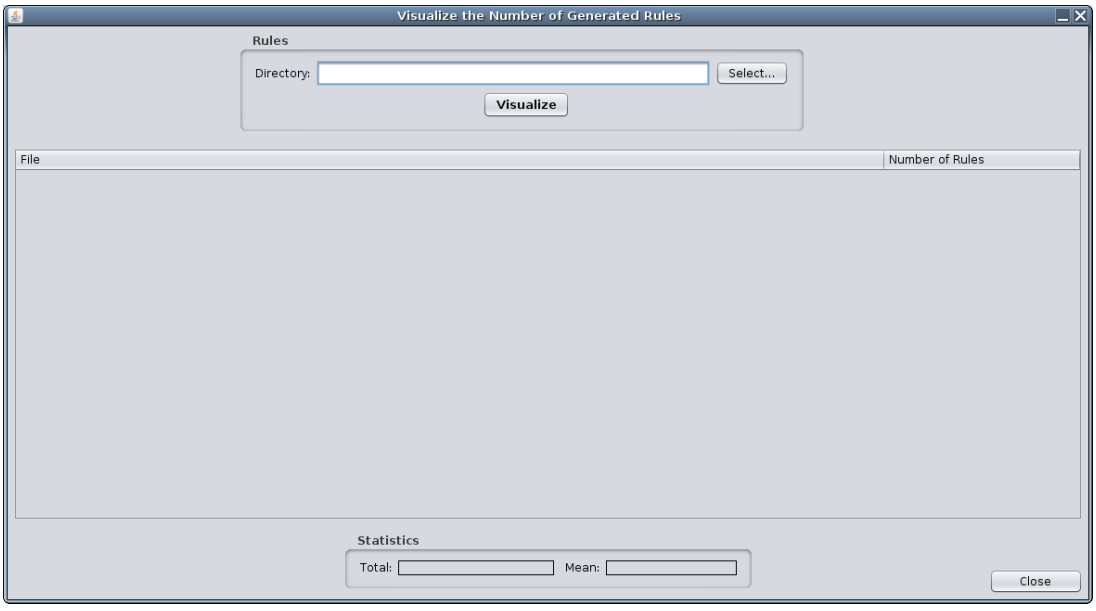


Figura 3.6: Tela para visualizar o número de regras geradas para cada documento da coleção.

Esta funcionalidade é útil para o usuário verificar o número de regras que estão sendo

geradas e com isso calibrar o valor de suporte mínimo utilizado para aumentar ou diminuir o número de atributos.

3.2.2 Comparando os atributos de diferentes representações

A utilização de diferentes formas de mapeamento e das diferentes medidas de interesse objetivas empregadas para a geração e seleção de regras de associação, podem fazer com que sejam gerados diferentes atributos. Para que se possam verificar os diferentes atributos que duas representações geradas, e com isso escolher a representação apropriada para a tarefa de interesse do usuário, foi desenvolvida uma funcionalidade que permite comparar os atributos gerados por diferentes representações *bag-of-related-words*.

Na Figura 3.7 é apresentada a tela que permite comparar os atributos das representações. É necessário informar dois arquivos no formato "`discover.names`" das representações que se deseja comparar nas caixas `File 1` e `File 2`. Após clicar no botão `Analyse` são exibidos o número de atributos de cada uma das representação, o número de atributos iguais e o número de atributos diferentes do arquivo em `File 2` em relação ao arquivo em `File 1`.

Também é possível salvar a lista de atributos iguais ou diferentes clicando no botão `Save` abaixo de cada lista.

3.2.3 Contando o número de atributos compostos por n palavras

A número de palavras que compõem os atributos da representação *bag-of-related-words* é aumentado enquanto as palavras possuam um suporte acima de um limiar e que as relações entre elas apresentam também um valor maior que um limiar informado. Foi desenvolvida uma funcionalidade na ferramenta FEATuRE para que se possa verificar a quantidade de palavras que compõem os atributos da representação *bag-of-related-words*, apresentada na Figura 3.8.

Nessa tela, são apresentado o número de atributos compostos por uma, duas, até n palavras. É necessário informar o caminho de um arquivo no formato "`discover.names`" na caixa `Path` e acessar o botão `Count`. Após o processamento, é exibida uma lista contendo o número de palavras por atributo (`Number of Words in the Feature`) com o respectivo número de atributos composto por aquela quantidade de palavras (`Number of Features`).

3.2.4 Convertendo a representação *bag-of-related-words* para o formato `arff`

A ferramenta WEKA (*Waikato Environment for Knowledge Analysis*) contém uma das ferramentas mais utilizadas para Mineração de Dados e Textos (Witten e Frank, 2005). Ela

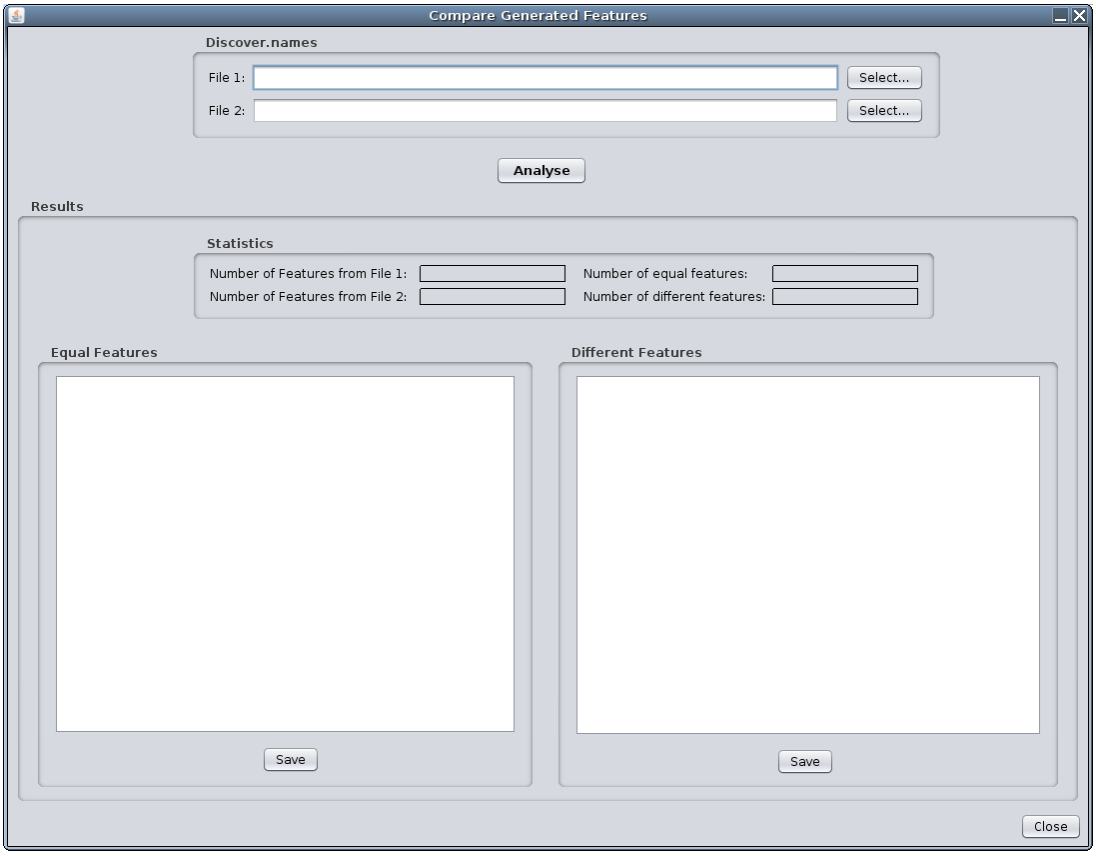


Figura 3.7: Tela para compara os atributos gerados por duas representações diferentes.

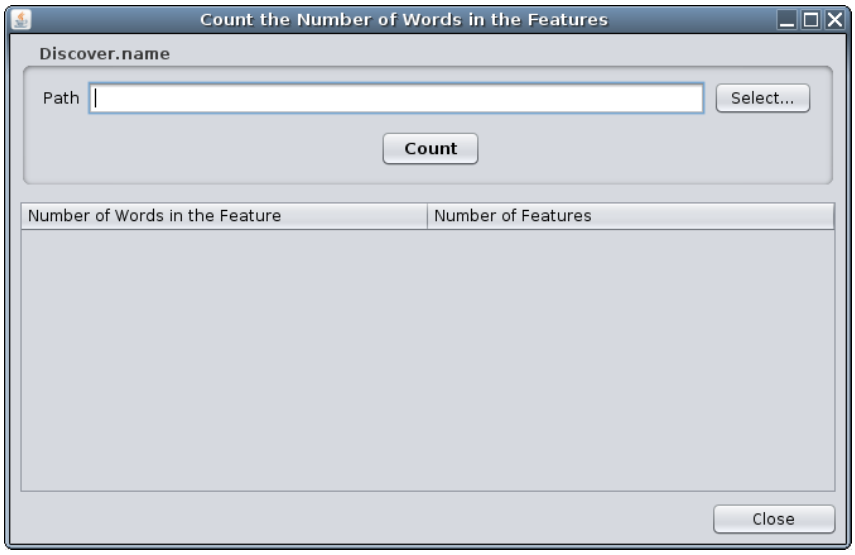


Figura 3.8: Tela para contar o número de atributos compostos por n palavras.

apresenta uma série de algoritmos de aprendizado de máquina e métodos de classificação, agrupamento, extração de regras de associação, seleção de atributos, além de uma série de funcionalidades para pré-processamento e visualização de dados. O `arff` é o principal formato de entrada para esta ferramenta. Na ferramenta FEATuRE foi desenvolvida uma

funcionalidade para converter a representação *bag-of-related-words* para o formato `arff`.

Na Figura 3.9 é apresentada a tela para conversão da representação *bag-of-related-words* para o formato `arff`. É necessário informar o diretório contendo os arquivos no formato "`discover.data`" e "`discover.names`". Caso o diretório informado contenha uma série de outros diretórios com arquivos `.data` e `.names`, pode-se utilizar o nome dos diretórios como nome do arquivo `arff` resultante selecionando a opção `Use the directory name as file name`. O usuário pode especificar manualmente o nome do arquivo `.arff` resultante do processo selecionando a opção `Specify the file name` e digitar o nome desejado na caixa `File name`. Caso a coleção possua rótulos, deve-se selecionar a opção `The documents have classes`.

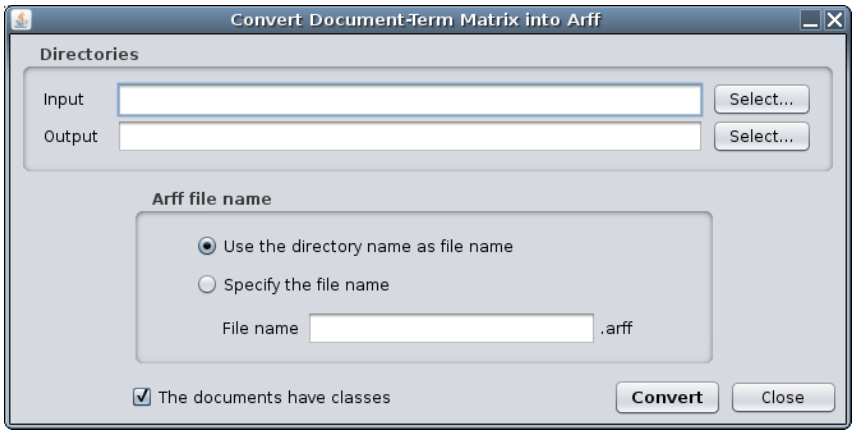


Figura 3.9: Tela para converter a representação *bag-of-related-words* para o formato `arff`.

3.3 Classes da ferramenta FEATuRE

A ferramenta FEATuRE foi implementada na linguagem de programação java. Ao todo foram criadas 31 classes. Na Figura 3.10 são apresentadas as principais classes da ferramenta FEATuRE e a interação entre elas. A seguir serão descritas as principais classes.

- **Menu.java**: menu principal da ferramenta FEATuRE. Responsável por disparar todas as outras funcionalidades da ferramenta.
- **ConverterATAT**: converte arquivos contendo texto em arquivos de transações.
- **ConverterATAR**: converte arquivos de transações em arquivos de regras. Para a geração das regras é utilizado o arquivo `aprioriwin.exe`. Este arquivo é a implementação do algoritmo Apriori desenvolvido por Chistian Borguelt[2]. Este arquivo

[2]http://www.borgelt.net/apriori.html

é uma compilação para o sistema operacional Windows. Caso a ferramenta esteja sendo executada no linux é necessário ter o aplicativo `Wine` instalado.

- **ConververARAA**: converte arquivos de regras em arquivos de atributos.
- **ConverterAAMAV**: converte arquivos de regras em arquivos no formato "`discover.data`" e "discover.names" como apresentado na Seção 3.1.4.
- **CompararATR**: compara os atributos de duas representações diferentes.
- **VisNumRegras**: obtém o número de regras geradas para cada documento de uma coleção.
- **ConverterDiscArff**: gera um arquivo no formato `arff` utilizando arquivos no formato "`discover.data`" e "`discover.names`".
- **ContarAtributos**: conta quantos atributos são composto por uma, duas, até n palavras, onde n é o número máximo de palavras que compõe um atributos em uma dada representação.
- **GerenciadorArquivos**: retorna os arquivos dentro de diretórios e os diretórios de uma coleção de documentos textuais.
- **TextToTransaction**: realiza o pré-processamento e o mapeamento dos documentos textuais em arquivos de transações.
- **Cleaner**: elimina caracteres que não serão utilizados no decorrer do processo de mineração de textos e a padronização de caixa para caixa baixa. Os caracteres permitidos são: `a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`, `i`, `j`, `k`, `l`, `m`, `n`, `o`, `p`, `q`, `r`, `s`, `t`, `u`, `v`, `w`, `x`, `y`, `z`, `.`, `!`, `_` e `?`.
- **StemPt**: radicalização das palavras para a língua portuguesa. Para a radicalização em português foi utilizada a biblioteca Java `ptstemmer` disponível em `http://code.google.com/p/ptstemmer/`.
- **StemEn**: radicalização das palavras para a língua inglesa. Para a radicalização em inglês foi utilizada a classe Java disponível em `http://tartarus.org/~martin/PorterStemmer/java.txt`.
- **removeStopwords**: remova as *stopwords*. As *stopwords* estão contidas no arquivos `stopPort.txt` e `stopIngl.txt` para as línguas portuguesa e inglesa respectivamente. Cada linha desses arquivos corresponde a uma `stopword`.
- **Transactions**: mapeamento do texto pré-processado em transações.

- **Confianca**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Confiança.
- **Gini**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Gini Index.
- **ItemsetSimples**: geração dos *itemsets* para serem utilizados como atributos,
- **MutualInformation**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Informação Mútua.
- **JMeasure**: cálculo, *ranking* e poda das regras utilizando a medida objetiva J-Measure.
- **Kappa**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Kappa.
- **Lift**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Lift.
- **QYule**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Yule's Q.
- **LinearCorrelation**: cálculo, *ranking* e poda das regras utilizando a medida objetiva Índice de Correlação Linear.
- **Maximal**: geração dos *itemsets* frequêntes máximos para serem utilizados como atributos.

O código-fonte da ferramenta FEATuRE encontra-se disponível em http://sites.labic.icmc.usp.br/feature/codigo-fonte/.

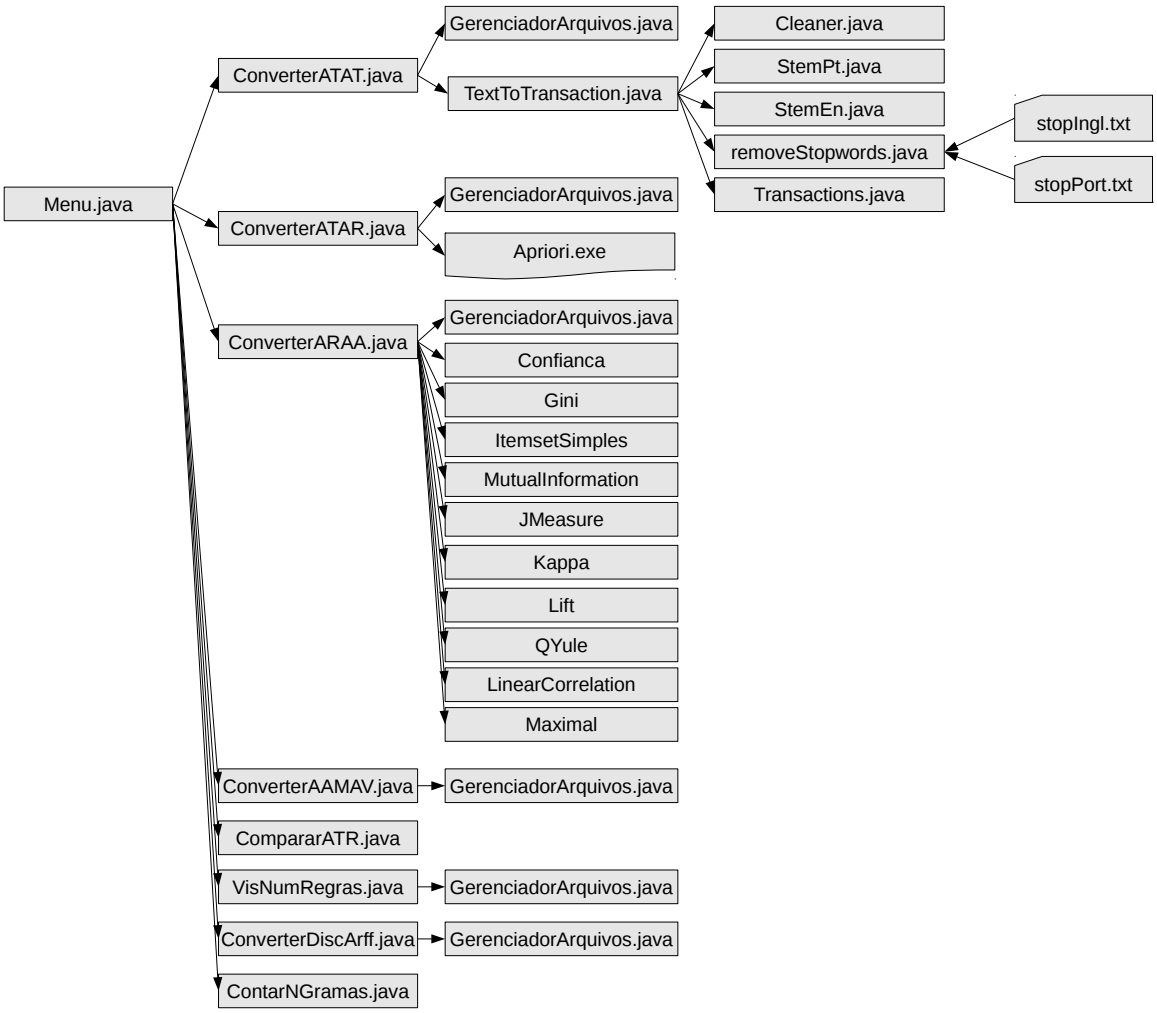


Figura 3.10: FEATuRE - Tela inicial.

Capítulo 4

Considerações Finais

Técnicas de Mineração de Textos têm-se tornado cada vez mais importantes para a organização e extração de conhecimentos devido à grande quantidade de documentos textuais disponíveis. Porém, os documentos textuais geralmente encontram-se em um formato não estruturado ou semi-estruturado, ou seja, não estão em um formato apropriado para a extração de padrões. Portanto, é necessária a estruturação dos documentos textuais de uma coleção para que as técnicas de Mineração de Textos possam ser aplicadas.

Em (Rossi e Rezende, 2011a,b) é apresentada a representação *bag-of-related-words*, criada para gerar uma representação de coleções de documentos textuais cujos atributos sejam compostos por palavras relacionadas e produzindo resultados de mais fácil entendimento ao usuário do processo de Mineração de Textos. Além disso, essa representação evita uma série de problemas encontrados na literatura quando se utilizam atributos compostos por mais de uma palavra. Dentre eles, podemos citar a alta dimensionalidade, e com isso a necessidade da aplicação de métodos de seleção de atributos, a necessidade da análise da toda a coleção de documentos para a geração de atributos, e a geração de atributos com pouco significado ao usuário.

Neste relatório técnico foi apresentada a ferramenta FEATuRE (Features gEnerator based on AssociaTion RulEs) que implementa todos os passos da geração da representação *bag-of-related-words*. Foram apresentadas todas as telas da ferramenta, os arquivos de entrada e saída, bem como os utilitários também implementados.

Referências Bibliográficas

Agrawal, R. e R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *VLDB'94: Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, USA, pp. 487–499. Morgan Kaufmann Publishers Inc. Citado na página 5.

Ahonen-Myka, H., O. Heinonen, M. Klemettinen, e A. I. Verkamo (1999). Finding co-occurring text phrases by combining sequence and frequent set discovery. In *Proceedings of the Workshop on Text Mining: Foundations, Techniques and Applications. Em conjunto com o IJCAI'99.*, pp. 1–9. Citado na página 3.

Bekkerman, R. e J. Allan (2004). Using bigrams in text categorization. Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst. Citado na página 3.

Blanchard, J., F. Guillet, R. Gras, e H. Briand (2005). Using information-theoretic measures to assess association rule interestingness. In *ICDM'05: Proceedings of the International Conference on Data Mining*, pp. 66–73. Citado na página 14.

Borgelt, C. (2004). Implementação do algoritmo apriori (versão 4.27) - `http://www.borgelt.net/apriori.html`. Citado na página 19.

Carvalho, A. L. C., E. S. Moura, e P. Calado (2010). Using statistical features to find phrasal terms in text collections. *Journal of Information and Data Management 1*(3), 583–597. Citado na página 3.

Carvalho, V. R. e W. W. Cohen (2006). Improving "email speech acts" analysis via n-gram selection. In *ACTS '09: Proceedings of the Workshop on Analyzing Conversations in Text and Speech*, Morristown, NJ, USA, pp. 35–41. Association for Computational Linguistics. Citado na página 3.

Ebecken, N. F. F., M. C. S. Lopes, e M. C. A. Costa (2003). Mineração de Textos. In S. O. Rezende (Ed.), *Sistemas Inteligentes: Fundamentos e Aplicações* (1 ed.)., Chapter 13, pp. 337–370. Manole. Citado na página 1.

Fagan, J. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science 40*(2), 115–132. Citado na página 3.

Feldman, R. e J. Sanger (2006, December). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. Citado na página 1.

Fürnkranz, J. (1998). A study using n-gram features for text categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence. Citado na página 3.

Gantz, J. F. e D. Reinsel (2010). The digital universe decade – are you ready? *External Publication of IDC (Analyse the Future) Information and Data*, 1–16. Citado na página 1.

Geng, L. e H. J. Hamilton (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys 38*(3), 9. Citado na página 14.

Guillet, F. e H. J. Hamilton (Eds.) (2007). *Quality Measures in Data Mining*, Volume 43 of *Studies in Computational Intelligence*. Springer. Citado na página 14.

Gupta, V. e G. Lehal (2009). A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence 1*(1), 60 – 76. Citado na página 1.

McNicholas, P. D., T. B. Murphy, e M. O'Regan (2008). Standardising the lift of an association rule. *Computational Statistics & Data Analysis 52*(10), 4712–4721. Citado na página 21.

Mladenic, D. e M. Grobelnik (1998). Word sequences as features in text-learning. In *ERK'98: Proceeding of the Electrotechnical and Computer Science Conference*, pp. 145–148. Citado na página 3.

Rossi, R. G. (2011). Representação de coleções de documentos textuais por meio de regras de associação. Master's thesis, Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação. Citado nas páginas 3 e 11.

Rossi, R. G. e S. O. Rezende (2011a). Building a topic hierarchy using the bag-of-related-words representation. In *ACM DOCENG'2011: ACM Symposium on Document Engineering*. Citado nas páginas 3, 5, e 33.

Rossi, R. G. e S. O. Rezende (2011b). Generating features from textual documents through association rules. In *ENIA'2011: Encontro Nacional de Inteligência Artificial.*, pp. 311 – 321. Citado nas páginas 3, 5, e 33.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. Citado na página 1.

Schenker, A., M. Last, H. Bunke, e A. Kandel (2003). Classification of web documents using a graph model. In *ICDAR'03: Proceedings of the International Conference on Document Analysis and Recognition*, Washington, DC, USA, pp. 240. IEEE Computer Society. Citado na página 1.

Scott, S. e S. Matwin (1999). Feature engineering for text classification. In *ICML'99: Proceedings of the International Conference on Machine Learning*, San Francisco, CA, USA, pp. 379–388. Morgan Kaufmann Publishers Inc. Citado na página 2.

Song, M. e Y. fang Brook Wu (2008, September). *Handbook of Research on Text and Web Mining Technologies* (illustrated edition ed.). Information Science Reference. Citado na página 1.

Tan, P.-N., V. Kumar, e J. Srivastava (2002). Selecting the right interestingness measure for association patterns. In *SIGKDD'2002: Proceedings of the International Conferenceon Knowledge Discovery and Data Mining*, pp. 32–41. ACM. Citado na página 14.

Tesar, R., V. Strnad, K. Jezek, e M. Poesio (2006). Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In *DocEng'06: Symposium on Document Engineering*, pp. 138–146. Citado na página 3.

Witten, I. H. e E. Frank (2005). *Data Mining: Practical machine learning tools and techniques* (2 ed.). Morgan Kaufmann. Citado nas páginas 24 e 26.

Wu, J., H. Xiong, e J. Chen (2009). Adapting the right measures for k-means clustering. In *SIGKDD'09: Proceeding of the International Conference on Knowledge Discovery and Data Mining*, pp. 877–886. ACM. Citado nas páginas 2 e 14.

Yang, Z., L. Zhang, J. Yan, e Z. Li (2003). Using association features to enhance the performance of naïve bayes text classifier. In *ICCIMA '03: Proceeding of the International Conference on Computational Intelligence and Multimedia Applications*, pp. 336. IEEE Computer Society. Citado na página 3.

Zhang, X. e X. Zhu (2007). A new type of feature - loose n-gram feature in text categorization. In *IbPRIA'07: Proceeding of the Iberian Conference on Pattern Recognition and Image Analysis*, pp. 378–385. Springer. Citado na página 3.