

BLMSumm – Métodos de Busca Local e Metaheurísticas na Sumarização de Textos

Marcelo Arantes de Oliveira¹, Marcus Vinícius Guelpeli²

¹Centro Universitário de Barra Mansa (UBM)
Barra Mansa, RJ - Brasil

²Universidade Federal Fluminense (UFF)
Niterói, RJ - Brasil

marceloarantes19@gmail.br, mguelpeli@ic.uff.br

Abstract. *This paper describes the BLMSumm, a summarizer that can use local search methods and Metaheuristics to create a summary. The summarization problem is presented as optimization problem and a architecture to identification and choice of attributes, based on bipartite graphs, is shown in this work.*

Resumo. *Este artigo descreve o BLMSumm, um sumariador que tem por objetivo a utilização de diversos métodos de busca local e Metaheurísticas na construção de sumários. O problema de sumarização é apresentado como um problema de otimização e uma arquitetura para identificação e escolha de atributos utilizando grafos bipartidos também é apresentada aqui.*

1. Introdução

A sumarização é um processo que busca criar uma versão reduzida do texto original (Mark e Maybury 1999). Sumário é um resumo que tem o objetivo de encontrar a idéia principal do autor e representar esta em poucas linhas (Hutchins 1987). Sumários são textos reduzidos que transmitem as idéias principais e mais importantes de um texto original, sem perda da informatividade (Delgado *et al* 2010).

A taxonomia dos sumários pode ser dividida em: indicativos, informativos e sumários de críticas. Os sumários indicativos podem ser úteis para descobrir a idéia principal do autor mostrando-a ao leitor, caso haja interesse em aumentar o conhecimento sobre o tema, busca-se a leitura do texto original. Os sumários informativos possibilitam ter todas as informações, não havendo a necessidade de remeter-se ao texto original. Os sumários de críticas são comparativos (Hutchins 1987).

Processos automáticos de sumarização podem obter sumários (abordagem profunda) e extratos (abordagem superficial). Os sumários alteram a estrutura das frases originais reescrevendo-as com a finalidade de generalizar ou especificar as informações. Já os extratos seguem transposições das frases dos textos originais, e por algum tipo de método, escolhem as frases com maior relevância no texto e as colocam no extrato (Guelpeli *et al* 2008).

O maior problema da área de Sumarização Automática (SA) é gerar um sumário que não perca sua informatividade¹. Sendo assim a avaliação de SA é muito subjetiva, pois depende fundamentalmente da avaliação humana e não existe um consenso entre os avaliadores humanos frente aos resultados obtidos com os sumarizadores automáticos.

A representação do texto como um grafo bipartido e a representação do problema de sumarização automática como um problema de otimização combinatória são as inovações apresentadas e serão explicadas com detalhes em seções posteriores.

Este trabalho está organizado como segue. A Seção 2 apresenta modelo utilizado na aplicação dos métodos de melhoria iterativa em sumarização. A Seção 3 descreve a metodologia, o corpus utilizado, a forma de avaliação e os testes estatísticos. Na Seção 4 os experimentos e os resultados obtidos são descritos. Na Seção 5 o trabalho é concluído sugerindo trabalhos futuros.

2. Modelo BLMSumm

A idéia deste trabalho é apresentar um modelo que torne possível a utilização de heurísticas e metaheurísticas na solução do problema de geração de sumários automáticos, o BLMSumm. A Figura 1 apresenta o modelo desenvolvido, e seus detalhes são descritos nesta seção.

O BLMSumm, é composto por três fases: Pré-processamento, Processamento e Pós-Processamento.

2.1. Pré Processamento no BLMSumm

No BLMSumm, a fase de Pré-Processamento é executado em duas etapas: preparação e geração de um grafo bipartido que tem por finalidade fazer a identificar e seleção dos atributos como descrito na seção 2.1.2. O pré-processamento entrega a fase de Processamento uma lista de sentenças devidamente classificadas e “valoradas” a partir do grafo bipartido.

¹ Informatividade são as informações veiculadas através dos textos escritos ou visuais. A informatividade de um texto é medida de acordo com o conhecimento de mundo das pessoas a que ele se destina. Ou seja, um texto possui um alto grau de informatividade quando a compreensão mais ampla desse texto depender do repertório cultural do leitor. Um texto é mais informativo quanto menor for sua previsibilidade, e vice-versa. Para que haja sucesso na interação verbal, é preciso que a informatividade do texto seja adequada ao interlocutor.

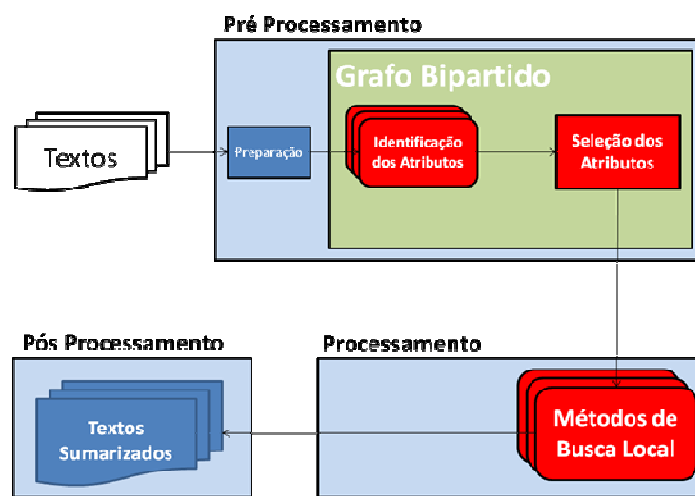


Figura 1: Modelo BLMSumm

2.1.1. Preparação

Foram utilizadas as técnicas de *case folding*, que consiste em transformar todas as letras para minúsculas (ou maiúsculas) e *stopwords*, que consiste na retirada de artigos, preposições e palavras pertencentes às classes abertas. Ao terminar a preparação do texto, um grafo bi-partido é gerado para auxiliar no processo de identificação e seleção dos atributos.

2.1.2. Grafo Bipartido

Um grafo $G=(V, E)$ consiste de um conjunto não vazio de vértices (V) e um conjunto de arestas (E), cada aresta sendo um conjunto contendo dois vértices de V . Um grafo Bipartido é o particionamento de V em dois subconjuntos V_1 e V_2 tais que qualquer aresta possui uma extremidade em V_1 e a outra em V_2 (Drozdek 2001).

No grafo bipartido gerado, o conjunto de vértices $V_1 \in V$ é dado pelo conjunto de sentenças do texto, enquanto o conjunto de vértices $V_2 \in V$ é dado pelo conjunto de palavras válidas do texto (palavras que não fazem parte do conjunto de *stopwords*). Uma aresta $E_x \in E$ é traçada no grafo para determinar que uma palavra faz parte de uma sentença. Considere o texto da Figura 2.

1. Luxemburgo rebate especulações.
2. Em momento algum existiu veto ao Adriano.
3. O problema de Adriano é na questão filosófica.
4. Luxemburgo afirma que momento não cabe a discussão sobre Adriano: “Não sei o por que dessa discussão no momento.”
5. Luxemburgo afirma que Adriano ainda pode ser contratado.

Figura 2: texto fictício baseado na reportagem de globoesporte.globo.com em 18/03/2011

O texto apresentado na Figura 2 gera o grafo bipartido apresentado na Figura 3, onde $V_1 = \{1, 2, 3, 4, 5\}$ e $V_2 = \{\text{Luxemburgo, rebate, especulações, momento, existiu, veto, Adriano, problema, questão, filosófica, afirma, cabe, discussão, sei, porque, pode,}$

ser, contratado}, as arestas são: $E = \{(1, \text{Luxemburgo}), (1, \text{rebate}), (1, \text{especulações}), (4, \text{Luxemburgo}), (2, \text{momento}), (2, \text{existiu}), \dots\}$.

Se uma palavra ocorre mais de uma vez em uma sentença, é adicionada a frequência desta ocorrência na aresta que a representa, como nas arestas: $\{(4, \text{discussão}), (4, \text{momento})\}$.

2.1.2.1. Identificação dos Atributos

O identificador de atributos do BLMSumm percorre o grafo bipartido determinando o valor de cada palavra e/ou sentença a partir de um cálculo de frequência previamente estabelecido. Neste trabalho foram utilizados dois tipos de cálculo de frequência de palavras:

1. Cada sentença terá o valor do somatório das frequências de suas palavras.
2. Cada sentença terá o valor do somatório do peso de suas palavras. As palavras da primeira sentença pesam dez vezes sua frequência e as demais possuem o peso coincidindo com sua frequência no texto.

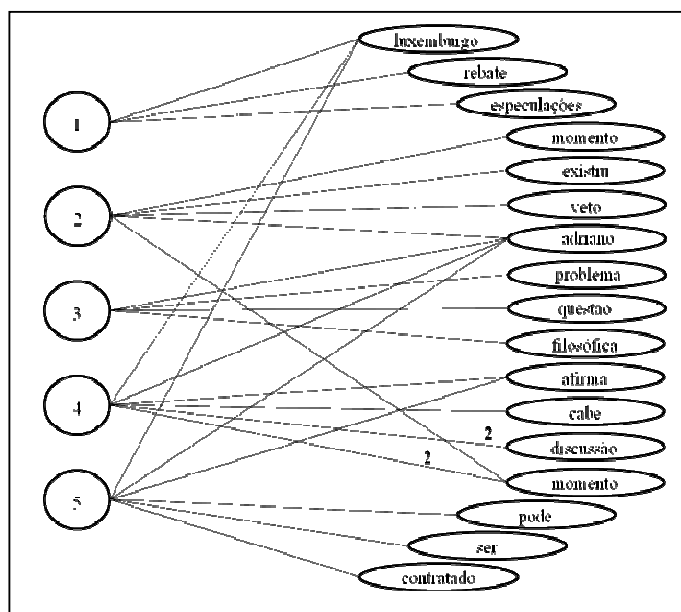


Figura 3: grafo bipartido que representa o texto: à esquerda V1 e à direita V2.

As tabelas, Tabela 1 e Tabela 2 apresentam a aplicação dos cálculos de frequência de palavras ao texto apresentado na Figura 2.

Tabela 1: Cálculo de Frequência 1 aplicada ao texto.

Sentença	Total de Palavras na Sentença	Lista de sentenças valoradas pelo cálculo de frequência 1
1	3	$3 + 1 + 1 = 5$
2	7	$3 + 1 + 1 + 4 = 9$
3	8	$1 + 4 + 1 + 1 = 7$
4	19	$3 + 2 + 3 + 3 + 1 + 2 + 2 + 4 = 20$
5	8	$3 + 2 + 4 + 1 + 1 + 1 = 12$

Tabela 2: Cálculo de Frequência 2 aplicada ao texto.

Sentença	Total de Palavras na Sentença	Lista de sentenças valoradas pelo cálculo de frequência 2
1	3	$30 + 10 + 10 = 50$
2	7	$3 + 1 + 1 + 4 = 9$
3	8	$1 + 4 + 1 + 1 = 7$
4	19	$30 + 2 + 3 + 3 + 1 + 2 + 2 + 4 = 47$
5	8	$30 + 2 + 4 + 1 + 1 + 1 = 39$

É possível utilizar outros cálculos de frequência que levem em consideração: a localização da palavra, a localização de sentenças, a relação entre as frequências e localização de sentenças e palavras do texto, Frequência Relativa, Termo de Frequência Inversa por Sentença (TF-ISF), *Ranking* pela frequência de termos (RTF), *Ranking* pela frequência de Sentença (RFS).

2.1.2.2. Seleção dos Atributos

Após a etapa de identificação do atributo começa a fase de seleção, esta fase caracteriza pela escolha de um subconjunto a partir de um conjunto. Este subconjunto permanece fiel a sua posição original melhorando o entendimento do modelo gerado (Liu e Yu 2005). Para um conjunto de dados que possui n atributos, existem $2^n - 1$ possibilidades de obter um subconjunto. Fica claro que na seleção de atributos existem cortes, que ocorrem mediante algum critério segundo (Dash e Liu 1997).

Na proposta deste trabalho não existe a criação de subconjuntos, mas sim uma lista de sentenças valoradas por algum cálculo de frequência adotado no item 2.1.2.1. A utilização de um grafo bipartido, para que a identificação e a seleção de atributos sejam feitas, proposta neste trabalho torna possível a utilização desta configuração.

2.1.3. Processamento no BLMSumm

Na fase de processamento tem a tarefa de escolher as sentenças que farão parte do sumário, entregando à fase pós processamento a lista de sentenças pertencentes ao sumário.

2.1.3.1. Sumarização como um problema de otimização

Considerando que cada sentença no sumário receberá um valor que determine a “importância” da mesma em função do texto, escolher o melhor sumário pode ser considerado como o processo de escolher o conjunto de sentenças de maior “importância” do texto, sem que a quantidade de palavras deste conjunto ultrapasse uma taxa de compressão. Neste caso, a sumarização superficial pode ser vista como um problema de otimização:

$$\text{Maximize } \sum_{i=1}^n h_i b_i \quad (1)$$

$$\text{Sujeito a } \sum_{i=1}^n p_i b_i \leq C \quad (2)$$

Onde:

- n – é o número total de sentenças em um texto.
- i – é a i -ésima sentença de um texto.
- p_i – é a quantidade de palavras da i -ésima sentença de um texto.
- h_i – é o valor da i -ésima sentença de um texto, dada pela função de avaliação.
- b_i – é um valor binário que determina se a i -ésima sentença de um texto, faz parte ($b_i = 1$) ou não faz parte ($b_i = 0$) de um sumário.
- C – é a quantidade máxima de palavras permitida a um sumário, que é dada em função da taxa de compressão.

Assim, cada estado é um sumário e pode ser visto como uma lista de sentenças, um número binário e/ou um número inteiro. Problemas

2.1.3.2. Métodos de Busca Local e Metaheurísticas

Algoritmos de busca local são construídos como uma forma de exploração do espaço de busca. Nestes um estado inicial é gerado por algum método (construtivo ou aleatório) e este é melhorado a cada iteração até que um estado ótimo seja atingido. Métodos de busca local tendem a ficar presos em ótimos locais, não gerando, conseqüentemente, uma solução ótima global para um determinado problema. As Metaheurísticas são desenvolvidas para fugirem destes ótimos locais e convergirem para uma solução ótima global.

2.1.3.2.1. Algoritmos Gulosos

Um algoritmo guloso sempre faz a escolha que parece ser a melhor no momento, ou seja, ele faz uma escolha ótima para as condições locais na esperança de que esta escolha leve a uma solução ótima para a situação global (Cormen e Leiserson 2002).

O algoritmo guloso escolhe a melhor solução localmente, sendo assim, a cada iteração o algoritmo seleciona a sentença de maior valor e, se o somatório de palavras contido no sumário atual adicionado à quantidade de palavras da sentença não ultrapassar à taxa de compressão, esta é adicionada.

Os cálculos de freqüência apresentados na seção 2.1.2.1., geram os valores das sentenças que, serão tratadas aqui como função de avaliação.

Considerando a taxa de compressão de 46%, ou seja, 24 palavras permitidas de um total de 45, e a função de avaliação 1 (cálculo de freqüência 1: seção 2.1.2.1) em associação com o algoritmo guloso escolheria as seguintes sentença:

1. Sentença 4: 19 palavras e 20 pontos;
 2. Sentença 1: 3 palavras e 5 pontos;
- Que totaliza de 22 palavras e 25 pontos (número binário correspondente: 01001).
Entretanto, outra combinação de sentenças poderia ser mais eficiente:
1. Sentença 2: 7 palavras e 9 pontos;
 2. Sentença 3: 8 palavras e 7 pontos;
 3. Sentença 5: 8 palavras e 12 pontos;
- Que totaliza de 23 palavras e 28 pontos (número binário correspondente 10110)

Esta solução não seria encontrada pelo algoritmo guloso utilizando a função de avaliação acima.

2.1.3.2.2. *Simulated Annealing*

A “localidade” empregada em algoritmos como o algoritmo Guloso, pode levar a soluções excelentes localmente, ou seja, a máximos locais que, em muitos casos, podem estar distantes da solução ótima para um determinado problema.

O *Simulated Annealing* é uma técnica de maximização / minimização que imita o processo de recozimento. A cada iteração o *Simulated Annealing* gera uma possível solução, se esta é a melhor do que a armazenada anteriormente, ela passa a ser a opção corrente. Existe a probabilidade de uma solução pior que a atual ser escolhida como corrente, com isto, o *Simulated Annealing* tenta fugir de máximos locais (Raynor 1999).

Cada sumário é representado por um número binário os bits válidos determinam que sentenças fazem parte do sumário. A cada iteração um número inteiro é escolhido aleatoriamente e convertido para binário, próximo sumário, este número é então associado à solução que ele representa, são verificadas as quantidades de palavras (se o número de palavras é maior que a taxa de compressão, a solução é descartada) e a pontuação do sumário (levando em consideração a função de avaliação). Se o próximo sumário tiver pontuação maior que o sumário corrente, então ele passa a ser o sumário corrente, caso contrário existe uma probabilidade de o próximo sumário se tornar o sumário corrente, esta probabilidade diminui com o tempo.

2.1.4. Pós Processamento no BLMSumm

O Pós-processamento, por sua vez, gera um arquivo texto sumarizado a partir das sentenças recebidas da fase de Processamento. Estes textos são os resultados dos algoritmos de sumarização. Para averiguar a grau de informatividade¹ foi utilizado o pacote ROUGE, que será descrito no item 3.3 deste trabalho

3. Metodologia

A contribuição deste trabalho é a de que métodos de busca local e metaheurísticas podem gerar sumários automáticos extrativos de qualidade similar aos sumarizadores presentes na literatura.

Para demonstrar esta contribuição, os métodos de busca foram implementados no BLMSumm e a seguinte metodologia de simulação foi empregada: foram usados corpus que são descritos no item 3.1 deste trabalho. Foram escolhidos sumarizadores que são detalhados no item 3.2. No aferimento da hipótese foi escolhido o pacote ROUGE que usa métricas *Recall*, *Precision* e *FMeasure* que são comumente usadas na mensuração do processo de sumarização e comentadas detalhadamente no item 3.3.

Foram definidos percentuais de compressão para serem usados nos sumarizadores. Cada texto foi sumarizado pelos algoritmos de sumarização com 50%, e 80%

3.1. Corpus

O corpus utilizado neste trabalho, foi o TeMário, com 100 textos extraídos de dois grandes jornais nacionais Folha de São Paulo e Jornal do Brasil. Eles têm as seguintes categorias: Especial, Mundo, Opinião da Folha de São Paulo e, Internacional e Política do Jornal do Brasil (Pardo e Rino 2004). Todos os textos possuem sumários de

referência criados por especialistas humanos. A estatística do corpus pode ser observada na Tabela 3.

Tabela 3. Mostra a estatística dos 100 textos do domínio jornalístico composto por 5 categorias e no idioma Português.

Itens	Nº de Palavras	Nº de Palavras + Numeral	Nº (%)	Caracteres	Caracteres + Espaços	Sentenças	Palavra por Sentença		Caracteres por Palavra	
Mínimos	424	431	0	2.160	2.649	14	424	14	2.160	431
Máximos	1.310	1.350	10,29	6.827	8.155	71	1.310	71	6.827	1.350
Totais	62.014	63.132	-	323.919	391.014	3.212	62.014	321.219,31	323.919	63.132
Médias	620,14	631,32	1,77	3.239,19	3.910,14	32,12	19,31		5,22	

3.2. Algoritmos de Sumarização

Para a simulação foram escolhidos sumrizadoros da literatura. Como critério para escolha dos algoritmos de sumarização destes experimentos foram definidos aqueles que tinham a possibilidade de definir percentuais de compressão por palavra, para isto, deveria possibilitar uma compressão de 50% e 80% do texto original.

3.2.1. Gist Average Keyword

No Gist_Average_Keyword (Pardo 2006) - a pontuação de sentenças pode ocorrer por um de dois métodos estatísticos simples: métodos *keywords* ou o método *average keywords*, isto é, o método *keywords* com normalização em função do tamanho das sentenças (medido em número de palavras) e depois ocorre o *ranqueamento* das sentenças em função da pontuação obtida, sendo que a sentença de maior pontuação é eleita *gist sentence*, isto é, a sentença que melhor representa a idéia principal do texto. Selecionam-se as sentenças que: (a) contenham pelo menos um *stem* em comum com a *gistsentence* selecionada na etapa anterior e (b) tenham uma pontuação maior do que um *threshold*, que é a média das pontuações das sentenças. Por (a), procura-se selecionar sentenças que complementem a idéia principal do texto; por (b), procura-se selecionar somente sentenças relevantes, Isto baseado no percentual de compressão determinado.

3.2.2. Gist Intrasentença

No Gist_Intrasentença (Pardo 2006) - é realizada em todas as sentenças pela exclusão das *stopwords* com a finalidade de redução da dimensionalidade.

3.2.3. Supor

O Supor (Módolo 2003) - seleciona para compor o extrato as sentenças que incluem as palavras mais frequentes do texto-fonte, pois elas representariam os conceitos mais importantes desse texto. A escolha de cada sentença é feita mediante a classificação de sua representatividade no texto. Para isso, é atribuído um escore a cada sentença, baseado na soma das frequências de suas palavras em todo o texto. Feito isso, determina-se um escore de corte (*threshold*) com base em medidas estatísticas e são, então, selecionadas as sentenças com as palavras mais frequentes do texto.

3.3. Avaliação dos sumários gerados

ROUGE (*Recall Oriented Understudy for Gisting Evaluation*) é, atualmente, uma das maneiras mais utilizadas para avaliar sumários automáticos é a medida (Lin e Hovy 2003). Basicamente, ROUGE é um pacote de avaliação de sumários, que surgiu para possibilitar a comparação direta entre um SA e o respectivo sumário humano usando como métrica n-gramas. Com essa ferramenta é possível analisar a proximidade de qualidade dos SA em relação aos sumários de referência humano. De uma forma geral, a ROUGE calcula o grau de informatividade dos extratos. ROUGE faz o cálculo baseando-se nos conjuntos de palavras em sequência (n-gramas) em comum entre os sumários de referência humanos e os Sumários Automáticos. Quanto maior o número de palavras em comum entre os sumários, maior será a nota dada ao sumário.

Para a validação dos resultados foram utilizados os testes estatísticos ANOVA de Friedman e coeficiente de concordância de Kendall que são empregados em conferências internacionais de avaliação de Sumários Automáticos como TAC (*Text Analysis Conference*), a mais relevante dentro da área de SA. Estes testes estatísticos verificam as diferenças de resultados existentes entre todas as amostras.

O teste ANOVA de Friedman é não paramétrico, seu objetivo é gerar uma ordenação entre as linhas observando o *ranking* de cada amostra (coluna), avalia-se, desta forma, se existe diferença significativa entre todas as amostras presentes no experimento.

O coeficiente de concordância de Kendall é não paramétrico e é utilizado para normalizar o ANOVA de Friedman. Sua finalidade é gerar um grau de concordância com o *rank* estabelecido. Quanto mais próximo de 0, menor é a concordância e quanto mais próximo de 1, maior é a concordância.

4. Experimentos

O BLMSumm foi executado utilizando os cálculos de frequência (funções de avaliação) apresentadas na seção 2.1.2.1. Para cada cálculo de frequência foram aplicados os métodos abaixo com os seus respectivos nomes detalhes e referência nos experimentos:

- **Guloso:** o Guloso é representado pelas referências gu1 e gu2 nos experimentos: (os números: 1 e 2 identificam que cálculo de frequência utilizados);
- **Guloso com Corte:** é uma variação do algoritmo guloso no qual as sentenças são ordenadas por seus respectivos valores e a cada iteração a sentença com maior valor heurístico é adicionada ao sumario sendo gerado. Ao encontrar uma sentença cuja quantidade de palavras adicionada às já pertencentes ao sumário sendo gerado ultrapasse a taxa de compressão estabelecida, o algoritmo é interrompido e o sumário é dado como gerado (excluindo a última sentença verificada). O **Guloso com Corte** é representado pelas referências guCorte1 e guCorte2 nos experimentos;
- **Simulated Annealing:** a agenda utilizada para possui 10000 posições tendo seus valores variando entre 0 e 373. O **Simulated Annealing** é representado pelas referências sa1 e sa2 nos experimentos;

4.1. Analise dos Resultados

A figura 4 apresenta os resultados produzidos pela avaliação baseando-se em sumários de referência humanos através do pacote ROUGE para compressão de 50%.

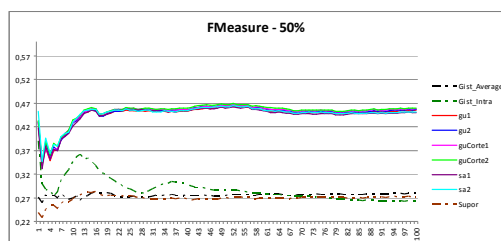


Figura 4: Resultados baseados na avaliação Rouge com compressão de 50%.

De acordo com a Figura 4, os métodos propostos neste trabalho obtiveram os melhores resultados. A pior avaliação foi do método Gist-Intrasentença. Os métodos da literatura, representados pelas linhas pontilhadas estão abaixo dos métodos propostos neste trabalho. Analisando a Figura 4, observa-se que não houve algoritmo com diferença significativa. Sendo assim, foram utilizados os testes estatísticos de ANOVA de Friedman e o coeficiente de concordância de Kendall para verificar a diferença entre as amostras. A Tabela 4 apresenta valores discordantes com a Figura 4 no método com o pior rendimento. Pelos testes estatísticos apresentados na Tabela 4 o pior desempenho foi atribuído ao método *Gist_Intrasentença* com 1,64 valor de ordem média. Conforme apresentado na Tabela 4 observa-se que os valores de ordem média estão a 65,5% dos 100% de concordância do coeficiente de Kendall do ranking estabelecido pelo teste estatístico ANOVA.

Tabela 4: Teste estatístico com Extratos Ideais.

Comparando amostras múltiplas relacionadas			
N	100	Graus de liberdade	8
qui-quadrado	524,7130376	p-nível	0
Coef. de concordância de Kendall	0,655891297	Ordem média	0,652415452
	Ordem média	Soma de ordens	Média
Gist_Average_Keyword	2,55	255	0,2807051
Gist_Intrasenteca	1,64	164	0,2623243
Guloso1	6,14	614	0,4551813
Guloso2	6,365	636,5	0,4567969
GulosoCorte1	6,94	694	0,4586951
GulosoCorte2	7,11	711	0,4598168
SimAnn1	5,93	593	0,450588
SimAnn2	6,175	617,5	0,45224
Suor	2.15	215	0.2721173

A Figura 5 apresenta os resultados produzidos pela avaliação baseando-se em sumários de referência humanos através do pacote ROUGE para compressão de 80%.

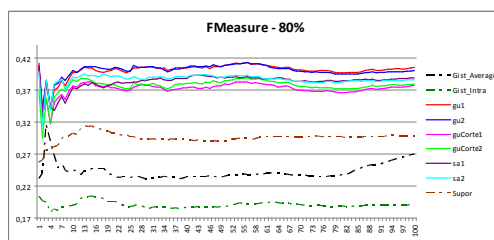


Figura 5: Resultados baseados na avaliação Rouge com compressão de 80%.

De acordo com a Figura 5, os métodos propostos neste trabalho obtiveram os novamente os melhores resultados.

Tabela 5: Teste estatístico com Extratos Ideais.

Comparando amostras múltiplas relacionadas			
N	100	Graus de liberdade	8
qui-quadrado	452,9488965	p-nível	0
Coef. de concordância de Kendall	0,566186121	Ordem média	0,561804162
	Ordem média	Soma de ordens	Média
Gist_Average_Keyword	2,845	284,5	0,2700962
Gist_Intrasenteca	1,125	112,5	0,1903291
Guloso1	7,24	724	0,405653
Guloso2	6,965	696,5	0,4008613
GulosoCorte1	5,415	541,5	0,377407
GulosoCorte2	5,51	551	0,3797337
SimAnn1	6,22	622	0,3895002
SimAnn2	6,11	611	0,3864529
Supor	3,57	357	0,2983482

Os métodos Gist Average, Intra e Supor, representados pelas linhas pontilhadas estão abaixo dos algoritmos propostos neste trabalho. Analisando a Figura 5, observa-se que não houve algoritmo com diferença significativa. Sendo assim, foram utilizados os testes estatísticos de ANOVA de Friedman e o coeficiente de concordância de Kendall para verificar a diferença entre as amostras. Conforme apresentado na Tabela 5 observa-se que os valores de ordem médios estão a 56,6% dos 100% de concordância do coeficiente de Kendall do *ranking* estabelecido pelo teste estatístico ANOVA. Estes valores justificam-se pela proximidade dos valores médios obtidos nos algoritmos propostos neste trabalho

5. Conclusão

Embora simples, o grafo bipartido mantém a estrutura de pontuação fora do escopo do método de solução empregado, isto permite testar diversos cálculos de frequência na identificação de atributos. Valorar as sentenças e representar o problema de sumarização como um problema de otimização possibilitam a utilização de diversos métodos de busca presentes na literatura.

O BLMSumm trabalha com *stopwords* e para utilizá-lo em outro idioma, a lista de *stopwords* deste deve ser fornecida. A escolha do método de busca é feita pelo usuário. Nenhum algoritmo de treinamento é utilizado.

Apesar do não conhecimento do melhor sumário automático extrativo classificado pelo ROUGE em cada texto do corpus TeMário, constatou-se que o BLMSumm é competitivo com relação aos métodos presentes na literatura para este domínio, como visto nas Figuras 4 e 5.

5.1. Trabalhos Futuros

Utilizar o grafo bipartido para gerar diversas simulações com diferentes cálculos de frequências, variar métodos de busca local, implementar métodos de busca populacional, variar domínios e idiomas, testar em diversas compressões.

Encontrar os melhores sumários, a partir do corpus Temário, classificados pelo ROUGE.

6. Referências Bibliográficas

- Cormen, T. H., and Leiserson C. E. (2002) “Algoritmos - Teoria e Prática.” *Segunda Edição*. Rio de Janeiro, RJ: Campus.
- Dash, M., and Liu H. (1997) "Feature Selection for Classification." *Intelligence Data Analise*, no. 1 131-136.
- Delgado, C. H., Vianna C. E., and Guelpeli, M. V. C.(2010) "Comparando sumários de referência humano com extratos ideais no processo de avaliação de sumários extrativos." *IADIS Ibero-Americana WWW/Internet*, 293-300.
- Drozdek, A. (2001) “Data Structures and Algorithms in C++”, *2nd edition*. Pacific Grove, CA - USA: Brooks / Cole.
- Guelpeli, M. V. C., Bernardini F. C., and Garcia A. C. B. (2008) "Todas as Palavras da Sentença como Métrica para um Sumarizador Automático."
- Hutchins, J. (1987) "Summarization: some problems and methods." *The frontier of informatics*, : 151-173.
- Lin, C., and Hovy, E. H.(2003) "Automatic Evaluation of Summaries Using N-gram." *Proceedings of the Language Technology Conference*.
- Liu, C. Y., and Yu L. (2005) "Toward Integrating Feature Selection Algorithms for Classification and Clustering." *IEEE Transactions on Knowledge and Data Engineering*.
- Mark, I., and Maybury M. T. (1999) “Advances in automatic text summarization.” Cambridge, USA: MIT Press.
- Módolo, M. (2003) "SuPor: um Ambiente para a Exploração de Métodos Extrativos para a Sumarização Automática de Textos em Português." *Dissertação de Mestrado - UFSCar*.
- Pardo, T. A. S. (2006) "Estrutura textual e multiplicidade de tópicos na sumarização automática: o caso do sistema GistiSumm." *Relatório Técnico (NILC - ICMC - USP)*.
- Pardo, T. A. S., and Rino, L. H. M. (2004) "TeMário: Um Corpus para Sumarização Automática de Textos." *Relatório Técnico (NILC - ICMC - USP)*.
- Raynor, W. (1999) “The International Dictionary of Artificial Intelligence.” New York, NY - USA: Amacon.