UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE TECNOLOGIA E GEOCIÊNCIAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LUCIANO DE SOUZA CABRAL

Uma Plataforma para Sumarização Automática de Textos Independente de Idioma.

ORIENTADOR: Prof. Dr. Rafael Dueire Lins

CO-ORIENTADOR: Prof. Dr. Frederico Luiz Gonçalves de Freitas

Recife/2015

LUCIANO DE SOUZA CABRAL

Uma Plataforma para Sumarização Automática de Textos Independente de Idioma.

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de Doutor em Engenharia Elétrica /Telecomunicações.

Orientador: Prof. Dr. Rafael Dueire Lins
Co-orientador: Prof. Dr. Frederico Luiz G. de Freitas

Recife/2015

Catalogação na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

C117p Cabral, Luciano de Souza.
Uma plataforma para sumarização automática de textos independente de idioma / Luciano de Souza Cabral. - Recife: O Autor, 2015.
138 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Rafael Dueire Lins.
Coorientador: Prof. Dr. Frederico Luiz Gonçalves de Freitas
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2015.
Inclui Referências e Apêndices.

1. Engenharia Elétrica. 2. Inteligência Artificial. 3. Processamento de Linguagem Natural. 4. Sumarização. 5. Tradução. 6. Análise de textos web. I. Lins, Rafael Dueire. (Orientador). II. Freitas, Frederico Luiz Gonçalves de. III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2015-81

Universidade Federal de Pernambuco
Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE

TESE DE DOUTORADO

LUCIANO DE SOUZA CABRAL

TÍTULO:

"UMA PLATAFORMA PARA SUMARIZAÇÃO AUTOMÁTICA DE TEXTOS INDEPENDENTE DE IDIOMA"

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, CIN/UFPE; FREDERICO LUIZ GONÇALVES DE FREITAS, CIN/UFPE; STEVEN JOHN SIMSKE, HEWLETT PACKARD/USA; BERNARD ESPINASSE, UNIVERSITÉ AIX DE MARSAILLE/FRANCE; VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE e ALUÍZIO FAUSTO RIBEIRO ARAÚJO, CIN /UFPE, sob a presidência do primeiro, consideram o candidato **Luciano de Souza Cabral** **APROVADO**.

Recife, 27 de Fevereiro de 2015.

CECÍLIO JOSÉ LINS PIMENTEL Coordenador do PPGEE	**RAFAEL DUEIRE LINS** Orientador e Membro Titular Interno
STEVEN JOHN SIMSKE Membro Titular Externo	**VALDEMAR CARDOSO DA ROCHA JÚNIOR** Membro Titular Interno
BERNARD ESPINASSE Membro Titular Externo	**ALUÍZIO FAUSTO RIBEIRO ARAÚJO** Membro Titular Externo

FREDERICO LUIZ GONÇALVES DE FREITAS
Co-orientador

Aos meus avós, pela sabedoria que a experiência os proporcionou.

Aos meus pais, pelo esforço e firmeza frente às dificuldades.

Aos orientadores Rafael Lins e Fred Freitas pela atenção, compreensão e aprendizado.

A minha irmã Shirley e o sobrinho Luan, pelo carinho e alegria atribuídos.

À minha esposa Laís e ao filho Lucca, pelo amor, paciência e compreensão.

A todos os meus familiares e amigos pela ajuda e prontidão.

Agradecimentos

Agradeço a meus pais, Severo e Lucy Cabral, e aos meus avôs maternos e paternos, Manoel e Maria Souza (*in memorian*), José Belém (*in memorian*) e Marina Cabral (*in memorian*), pois ao longo dos anos possibilitaram meu amplo aprendizado de vida. À minha namorada que tornou-se esposa no decorrer do curso, Laís e sua família, pela compreensão, carinho e amor cedidos neste período, assim como o fruto da nossa união, o mais novo membro da sociedade Pernambucana, Lucca. Aos meus orientadores, Rafael Lins e Fred Freitas, por toda atenção, amizade, paciência e dedicação prestadas durante todas as Pós-Graduações, além da Profa. Sandra Siebra, pela atenção e por todas as chances na área acadêmica a mim proporcionadas na graduação. Pelos conhecimentos adquiridos nas disciplinas cursadas, agradeço aos professores Hélio Oliveira, Valdemar Júnior, Fernando Campello, Rafael Lins, Fernanda Alencar, dentre outros. E aos caros Eduardo, Gabriel, Rafael, Rinaldo, Bruno, Hilário e Jamilson, que se tornaram amigos, unidos pelo acaso, juntos na pós-graduação, cujo apoio foi fundamental para adaptação, desenvolvimento, pesquisa e aquisição dos conhecimentos necessários, além do stress compartilhado. Ao Instituto Federal de Pernambuco, departamento de educação a distância da UFRPE e Faculdade Maurício de Nassau, pelo incentivo à pesquisa. Além da UFPE e do DES, pelo acolhimento e excelente formação acadêmica, a mim concedidos. Aos amigos, pela torcida e ajuda nas diversas árduas batalhas enfrentadas durante o período da pós-graduação, de fato, não foi nada fácil.

“Pede, e dar-se-vos-á; buscai e achareis; batei, e abrir-se-vos-á.

Pois todo o que pede recebe; o que busca encontra; e, a quem bate, abrir-se-lhe-á.

(Mateus 7:7,8)

Resumo da Tese apresentada à UFPE como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

Uma Plataforma para Sumarização Automática de Textos Independente de Idioma.

Luciano de Souza Cabral

Fevereiro/2015

Orientador: Prof. Dr. Rafael Dueire Lins.
Co-orientador: Prof. Dr. Frederico Luiz Gonçalves de Freitas.

Área de Concentração: Telecomunicações.
Número de Páginas: 138.

A Sumarização Automática de Textos é o ramo da área de recuperação de informação que utiliza técnicas e algoritmos para identificar e coletar ou gerar sentenças relevantes a partir de documentos textuais. Claramente, o uso de Processamento de Linguagem Natural (PLN) revela-se benéfico ao processo de sumarização, principalmente quando se processam documentos sem nenhuma estrutura e/ou padrão definido. Dentre as variações do processo de sumarização, as técnicas extrativas são as mais bem estudadas até o momento, em sua maioria suportando o idioma inglês, com poucas variações de suporte a mais um idioma. A presente tese propõe uma plataforma de sumarização multi-idioma na qual, fornece 17 opções de algoritmos de sumarização, assim como a possibilidade de combinação dentre eles. Tais algoritmos são uma mescla de técnicas de sumarização extrativa utilizando modelos estatísticos (e.g. TF-IDF) e modelos linguísticos (PLN com WordNet). Além disso, a plataforma é 100% não-supervisionada, o que significa que não depende do ser humano em nenhuma parte de seu processamento, ainda possui um módulo de identificação de idiomas além de um processo de tradução intermediária, os quais provêm suporte a 25 idiomas até o momento. Os resultados obtidos nos experimentos sugerem que a plataforma apresenta bons níveis de sumarização com corpora relacionados com textos jornalísticos (CNN e Temário) em diferentes idiomas (Inglês, Espanhol e Português). Efetuando uma comparação com métodos conhecidos, e.g. SuPor e TextRank, a plataforma obteve 45% de

melhoria nos resultados para o corpus Temário no idioma português, se manteve dentre os melhores com o corpus CNN em inglês e resultados semelhantes com o corpus CNN em espanhol, no qual é novo e não possui resultados de outros sistemas até o momento. Além desses resultados, o seu tempo processamento é competitivo, atingindo-se em média 0,11 segundos por documento em inglês e 0,28 s para outras línguas. Desenvolvida em Java, a plataforma pode ser facilmente portável e reusada em pesquisas futuras, as quais podem ser direcionadas para preencher a lacuna da sumarização abstrativa, a qual é pouco explorada até o momento pela comunidade, tendo assim, muito a ser estudada e pesquisada.

Palavras-chave: Inteligência Artificial. Processamento de Linguagem Natural. Sumarização; Tradução. Análise de textos web.

Abstract of Thesis presented to UFPE as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

A language-independent platform for the automatic summarization of text documents.

Luciano de Souza Cabral

Fevereiro/2015

Supervisors: Prof. Dr. Rafael Dueire Lins;
Co-supervisor: Dr. Frederico Luiz Gonçalves de Freitas.
Area of Concentration: Telecommunications.
Number of Pages: 138.

Automatic Text Summarization is the branch of information retrieval that uses techniques and algorithms to identify, collect or generate relevant sentences from text documents. The use of Natural Language Processing (NLP) techniques has proven to be beneficial to the summarization process, especially when processing unstructured documents. Extractive summarization techniques are the best studied to date, mostly supporting the English language, with a few variations to support another language. This thesis proposes a multi-language summarization platform which implements 17 algorithms, as well as the possibility of combining them. Such extractive summarization techniques are based on statistical models (e.g. TF-IDF) or language models (e.g. N.L.P. with WordNet). Furthermore, the platform is 100% unsupervised, this means that processing does not need human interference. There is a module for language identification and an intermediate translation process, in which provides support to 25 languages, so far. The experimental results obtained suggest that the platform reached acceptable summarization levels tested on news text corpora (CNN and Temário) in English, Spanish and Portuguese. Comparing with known methods, e.g. SuPor and TextRank, the platform obtained an improvement of 45% in the results for the TeMário corpus in Portuguese language remained among the best in the CNN corpus in English and similar results with the CNN corpus in Spanish, which is new and not have results of competitors yet. In addition to these results, its processing time is competitive,

reaching an average of 0.11 seconds per document in English and 0.28 for the other languages tested. The platform was developed in Java, thus it is portable and can be easily reused in future research in abstractive summarization, a research area still little explored.

Keywords: Artificial Intelligence. Natural Language Processing. Language summarization. Machine translation. Web texts analysis.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

DIC	Dicionário
DOC	*Document* – extensão de arquivos de documento
EI	Extração de Informação
EUROPARL	*European Parliament*
GATE	*General Architecture for Text Engineering* (Arquitetura Geral para Engenharia Textual)
PDF	*Portable Document Format* (Formato de documento portável)
PLN	Processamento de Linguagem Natural
RI	Recuperação de Informação
RTF	*Rich Text Format* (Formato de texto enriquecido)
TXT	*Plain Text* – extensão de arquivos de texto
XML	*eXtensible Markup Language* (Linguagem de marcação extensível)

Sumário

1 Introdução

Só sei que nada sei.
Sócrates

Uma vida não questionada não merece ser vivida.
Platão

A Internet é hoje o maior repositório de informação da história da humanidade. A quantidade de informações disponíveis na *World Wide Web* cresce a cada ano num ritmo acelerado. Não é só grande em volume, mas também na diversidade, qualidade, confiabilidade das informações, e número de idiomas. A busca de estratégias automáticas que torne mais simples, confiável e eficiente a extração de informação torna-se cada dia mais importante. A sumarização automática de textos tem sido considerada como uma possível solução para o problema de volume, porque se destina a encontrar as informações relevantes em documentos, criando uma versão de tamanho menor do documento, que possibilita também uma ênfase em aspectos específicos do texto. Por exemplo, um texto que descreva um determinado medicamento pode ser visto de diversas perspectivas. Um paciente pode estar interessado em detalhes da posologia e horário mais conveniente da tomada do medicamento, o médico que o vai prescrever interessado em interações medicamentosas ou efeitos colaterais, um profissional de farmácia ou bioquímica pode buscar o princípio ativo, mecanismo de ação da droga.

Além da quantidade de dados, outros problemas são a falta de garantia de qualidade do conteúdo e a priorização do idioma das palavras chaves nos motores de busca da web, tornando-se cada vez mais difícil encontrar de forma eficiente uma informação útil, na ampla gama de documentos na web, até em conteúdos presentes em idiomas diferentes da pesquisa original. Por isso, é necessária a utilização de métodos para entender, classificar e apresentar, de forma clara e concisa, informações em qualquer idioma, permitindo assim ao usuário economizar recursos e tempo.

Uma possível solução para estes problemas é a sumarização multilíngue. Ela tem a responsabilidade de resumir o conteúdo seja qual for à língua original, domínio e gênero textual do documento. Sua principal função é fornecer uma versão minimizada do conteúdo, de modo que o usuário possa entender o significado do que está sendo transmitido pelo autor. Portanto, ajudaria a resolver os problemas supracitados, melhorando a análise e cobertura das buscas na web, por exemplo, incluindo documentos em diferentes idiomas, fornecendo um resumo eximindo o usuário da obrigatoriedade de leitura completa de um *site* ou documento para ter uma ideia exata do que está escrito.

Outros trabalhos encontrados na literatura tratam do problema da sumarização multilíngue. A referência (RADEV, *et al.*, 2004) apresenta 8 funcionalidades implementadas e suporte para duas línguas (chinês e inglês). O trabalho (EVANS, MCKEOWN, & KLAVANS, 2005) apresenta uma estratégia sumarização via similaridade de sentenças e agrupamento *(clustering)*, com suporte para o árabe e inglês. Roark e Fisher (2005) usam aprendizagem de máquina para obter uma classificação da sentença de forma supervisionada. Esse trabalho inclui um passo de tradução intermediária como proposto nesta tese, entretanto a obra não faz menção sobre a quantidade e quais os idiomas suportados. No trabalho (LITVAK, LAST, & FRIEDMAN, 2010), mais recentemente, usa algoritmos genéticos na tarefa de sumarização, também como alguns trabalhos anteriores, suporta apenas dois idiomas (inglês e hebraico), da mesma forma em Gupta (2013), que usa um algoritmo híbrido para a sumarização multilíngue, focando os idiomas Hindi e Punjabi.

Nesta tese, a independência de língua é atingida através da combinação de técnicas para identificação do idioma, tradução e um conjunto de técnicas de sumarização de texto dependentes e independentes de linguagem.

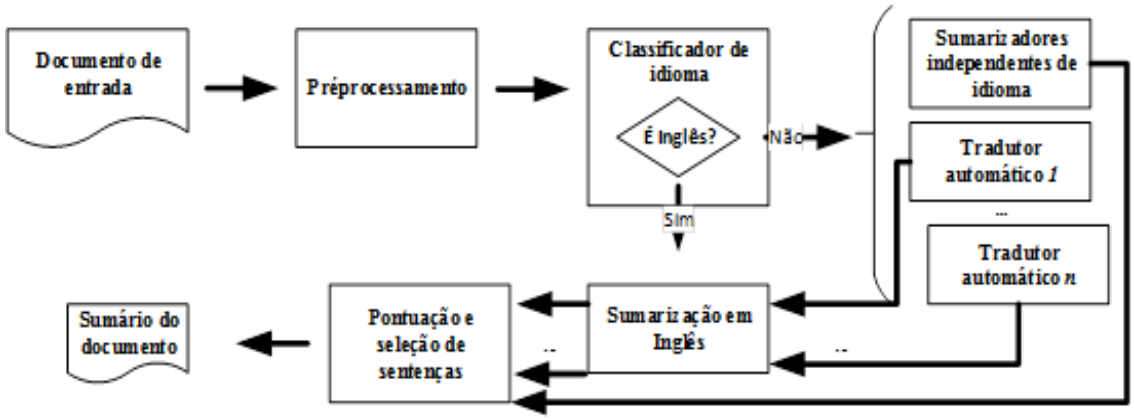

Figura 1. Arquitetura geral

As técnicas independentes de idioma, domínio e gênero textual são usadas diretamente no documento original. Para as técnicas dependentes de idioma, o texto original é automaticamente traduzido para o inglês. Existem hoje para a língua inglesa uma grande diversidade de ferramentas para análise léxica, sintática, semântica e prosódica de textos. A tradução do texto original para o inglês possibilita o uso de tais ferramentas. O texto original deve ser processado, removendo os elementos gráficos ou de multimídia que possam existir no mesmo. Parágrafos e sentenças são numeradas e então traduzidas individualmente, de forma que um mapeamento bijetivo associa cada sentença na língua original a uma sentença em inglês. Mais de um ambiente de tradução automática pode ser usado, permitindo a análise comparativa das traduções. É possível validar as sentenças contidas nos sumários gerados com as sentenças originais, evitando ruídos de tradução ou processamento, garantindo a sentença original intacta sendo exibida ao usuário pelo sumário gerado.

Para a avaliação do sistema aqui apresentado, foi necessário construir um corpus de alta qualidade uma vez que os corpora hoje existentes ou são pequenos ou não possuem toda a funcionalidade necessária. Portanto, um novo corpus foi criado composto de textos de notícias em diversas categorias, tais como: esportes, política, negócios, dentre outras. Os *sites* da CNN Internacional (inglês) e CNN México (espanhol), foram usados como fonte de dados e várias informações foram extraídas, tais como, os destaques (*highlights* ou *ló más importante*), o título do artigo, palavras-chave, entre outras, visando experimentos multilíngues, também utiliza-se o corpus TeMário (PARDO & RINO, 2003) fechar a avaliação utilizando 3 corpora com distintos idiomas.

As figuras 2 e 3 exemplificam quais são as regiões exploradas na página que contém a notícia para extração e armazenamento posterior. A primeira tratando-se do exemplo da página em inglês, a segunda em seguida, o que confirma a utilização do padrão, mesmo tratando-se de sites diferentes.

Figura 2. Identificação das regiões para extração do conteúdo.

Figura 3. Exemplo usando o site CNN México.

Para a avaliação da qualidade dos sumários obtidos foi desenvolvido um método de avaliação híbrido, que combina aspectos qualitativos e quantitativos, onde, selecionaram-se a partir de um grupo de notícias, sentenças que denotariam sumários de qualidade, tal procedimento foi efetuado de maneira supervisionada por um grupo de pesquisadores baseando-se nos destaques já disponibilizados pelos sites aos quais as versões do corpus

foram extraídas, chamados de padrão de referência (*gold standard*), assim, as sentenças geradas automaticamente pela plataforma proposta são comparadas ao padrão, de forma que os resultados mostrarão a quantidade de sentenças que coincidem com o *gold standard*, tendo-se um número grande, indicaria uma melhor qualidade, e o inverso é verdadeiro. O corpus TeMário também fornece sumários gerados por humanos, tornando-se compatível com os experimentos. A seguir, contextualiza-se a pesquisa com trabalhos recentes da área.

1.1 Contexto da pesquisa

Diante do exposto, os problemas existentes são árduos e sua solução se revela útil e benéfica, desde modo propõe-se uma plataforma de sumarização independente de idioma que faz uso de tarefas como identificação de idiomas, tradução e sumarização multilíngue, possibilitando ao usuário maior compreensão do conteúdo do documento procurado, despendendo um menor esforço administrativo, economizando tempo e recursos.

Sumarização de textos é o processo de criação automática de uma versão menor de um ou mais documentos textuais (FERREIRA, *et al.*, 2013). Essencialmente as técnicas de sumarização textual são classificadas em *extrativa* e *abstrativa* (LLORET & PALOMAR, 2012), onde na primeira são produzidos conjuntos de sentenças importantes de um documento, sem alterações. Já na segunda tenta-se melhorar a coerência do texto, eliminando redundâncias e deixando sentenças mais concisas, podendo-se produzir novas sentenças.

Como o mercado computacional tende à utilização de conteúdos de interesse, cada vez mais acessados via rede (*cloud*) e classificados (Web semântica ou Web 2.0) uma plataforma de sumarização independente de idioma, revela-se inovadora, possibilitando a criação de sumários de forma rápida seja online ou off-line, tais sumário podem ser utilizados em gama de estudos, por exemplo, utilização dos sumários como índices em engenhos de busca; possibilidade de classificação semântica do conteúdo; aplicações de técnicas de sumarização específicas de acordo com o texto de entrada; assim aproximando ainda mais o projeto dos tópicos mais atuais em computação. Atualmente a sumarização extrativa é a mais utilizada por ser menos complexa, e será o foco neste trabalho.

1.2 Aspecto Tecnológico e Inovador Principal da Tese

O aspecto tecnológico e inovador do trabalho realizado foi à criação de uma plataforma de sumarização independente de idioma, robusta, escalável e personalizável unindo conceitos identificação de idioma, tradução e sumarização. Em termos de estado da

arte, não se tem notícia de uma plataforma de sumarização que agregue estas características. Em adendo, vale salientar que na identificação é proposto um método híbrido inédito, denominado CALIM inspirado em três das técnicas mais consagradas na literatura (N-gramas, Classes fechadas e Perfis idiomáticos usando Frequência relativa; assim como é proposto um novo um método para avaliação de sumários que une aspectos qualitativos e quantitativos; o método de sumarização independente de idioma é baseado em algoritmos estatísticos e linguísticos, que sendo combinados podem alcançar resultados superiores aos concorrentes, i.e. TextRank. Validações individuais são efetuadas em cada módulo e ao final, tem-se a avaliação da plataforma com o ROUGE, o avaliador de sumários mais utilizado.

1.3 Desenvolvimento da Solução

Propõe-se neste trabalho uma solução composta no desenvolvimento de uma plataforma para sumarização independente de idiomas, com a responsabilidade de sumarizar o texto em diferentes línguas, domínios e gêneros textuais. Fornecendo assim uma pequena versão do texto, auxiliando no entendimento da mensagem textual pelo usuário. A solução é composta por funções intrínsecas, que no conjunto, retornam o resultado esperado, tais funções incluem a identificação do idioma, a tradução para língua inglesa, a sumarização do conteúdo e o fornecimento do resultado no idioma original. Desenvolvida em Java, a solução proposta agrega características de Portabilidade, Extensível, Robusta e Modular. Abaixo, tem-se a descrição individual da estratégia utilizada em cada módulo.

1.3.1 Identificação de idiomas

Para a identificação do idioma do documento, propõe-se a utilização de um algoritmo híbrido, baseado nos trabalhos de (DUNNING, 1994), (LINS & GONÇALVES, 2004) e (CAVNAR & TRENKLE, 1994). O algoritmo rotulado como CALIM é baseado em tabelas *hash* que são perfis idiomáticos, que leva em consideração a frequência relativa das janelas de caracteres (*n-gramas*) presentes em todas as línguas em estudo (21 no total) descrito em CABRAL *et al.* (2012). Experimentos individuais foram executados para validação da técnica e justificar a inclusão da estratégia supracitada na plataforma em referência.

1.3.2 Tradução de idiomas

No caso da tradução, utilizou-se a *Microsoft Translation API* (MICROSOFT, 2014) e *Google Translate API* (GOOGLE, 2012), tais ferramentas foram analisadas no experimento de identificação de idiomas, apesar de não serem compatíveis com a rapidez

dos outros algoritmos, apresentaram-se como uma solução viável e barata para tradução, uma vez que, a primeira é gratuita para solicitações de até 2.500 caracteres, bastando requerer apenas uma chave de registro para sua utilização no código fonte e a segunda, é paga, contém um limite de processamento estimado em dois milhões de caracteres por dia, podendo-se aumentar caso seja desejo do cliente, é paga e custa cerca de US$ 20.00 (vinte dólares americanos) a cada milhão de caracteres processados. Experimentos para validação do nível de influência dos tradutores na sumarização foram efetuados.

1.3.3 Sumarização

No módulo de sumarização, optou-se pela combinação de técnicas de sumarização, as quais provaram serem melhores de acordo com os experimentos de FERREIRA *et al.* (2013), são elas as técnicas baseadas em Frequência de Palavras e de Termos, Tamanho e Posição de Sentenças. Agregadas de forma combinada para acumular as pontuações obtidas por cada método e por fim, uma classificação ordenada decrescentemente, com limiar de seleção provido pelo usuário, que pode ser o número de sentenças ou um percentual do tamanho do texto original. O módulo contém no total, 17 técnicas diferentes implementadas, as quais podem ser utilizadas individualmente ou combinadas (a critério do usuário), adequando aos novos corpora que venham a ser utilizados e testados no futuro, visando melhores resultados.

1.4 Contribuições

Dentre as contribuições alcançadas neste trabalho, podem-se enumerar:

1. Desenvolvimento de uma plataforma para sumarização independente de idiomas, contendo:
    a. Um módulo de identificação de idiomas, retornando o idioma do documento original com bom custo benefício. Tal método de identificação foi inovador por combinar recursos de três técnicas de reconhecido sucesso na literatura, que geralmente eram usadas individualmente, obtendo-se como resultado uma identificação rápida e precisa, evidenciando nos experimentos o melhor custo benefício;
    b. Um módulo de tradução que efetua a tradução do texto original para a língua inglesa, requerida por alguns dos algoritmos de sumarização que necessitam de algum processo dependente de idioma, requer como entrada o idioma de origem e destino, além do conteúdo a ser traduzido;

c. Um módulo de sumarização que reúne 17 diferentes técnicas de sumarização, divididas em três grupos, que podem ser utilizadas individualmente ou combinadas a critério do usuário. Algumas são independentes de língua, tais como as técnicas baseadas em estatística como TF/IDF, já outras são dependentes do idioma, como *CuePhrases*, que usa um dicionário de palavras que sinalizam sentenças importantes no texto, para essas técnicas, os fluxos de identificação e tradução são utilizados, para as demais, a sumarização é efetuada de maneira direta;
d. A saída por padrão é fornecida no idioma do documento original, onde preventivamente usa-se um mapeamento bijetivo para assegurar que as sentenças do sumário gerado estejam idênticas às sentenças do texto original. Como existe um módulo tradutor integrado, há possibilidade de exibir o sumário gerado em um idioma especificado pelo usuário.

2. Para os experimentos foi necessário:
    a. A criação de um corpus de fácil entendimento e leitura, escolhendo-se notícias do portal CNN, tal corpus foi coletado, processado, revisado e obtido sumários abstrativos sugeridos pelos autores, chamados de *highlights*, chegando a 2000 documentos e em contínuo crescimento; Criaram-se sumários extrativos de referência (*gold standard*), escolhidos por um grupo de especialistas após analisarem uma porção das notícias presentes no corpus (cerca de 400), os quais foram classificados sumários de confiança para fins de avaliação; a escolha deste corpus é evidenciada pela facilidade de entendimento do conteúdo frente à concorrência;
    b. Na avaliação, tais sumários de qualidade *gold standard*, são utilizados para averiguar quais das técnicas de sumarização selecionadas conseguem coincidir suas sentenças de saída, com as contidas no *gold standard*, este resultado será proporcional ao número de coincidências. Tal método possui aspectos quantitativos e qualitativos (híbrido) que não são encontrados nos demais métodos de avaliação de sumários.
    c. Utilizou-se ainda a medida de avaliação consagrada pelos demais trabalhos da área, ROUGE, reforçando os resultados obtidos;
    d. Visando cumprir o experimento multilíngue, além do corpus em língua inglesa, construiu-se um corpus adicional em espanhol, baseado nas

notícias CNN México, que também possuem sumários gerados pelos autores, além de sumários extrativos gerados por especialistas. Para completar ainda mais o experimento foi utilizado o corpus em Português Brasileiro, TeMário (PARDO & RINO, 2003) no qual os textos são originalmente de notícias de jornais do Brasil, contendo sumários gerados automaticamente por sistemas e por especialistas.

3. Em particular, os experimentos mostram os seguintes resultados relevantes:
   - Resultados relativos à identificação de idiomas indicaram que o melhor custo benefício foi obtido pelo método aqui proposto chamado CALIM, apresentando resultados de alta precisão e baixo tempo de processamento:
     - 97,08% de precisão em 10s de processamento para 21 mil documentos *plain text* com tamanho médio de 172,90 bytes;
     - 99,99% de precisão em 2,776s de processamento para cerca de 60 mil documentos XML de tamanho médio de 84,51 KB (diferença explicada pelo tamanho médio de cada documento).
   - Resultados relativos ao módulo de tradução, a Microsoft API mostrou-se melhor que o Google, mantendo maior taxa de confiança de escolha de sentenças pelos sumarizadores após o processo de tradução, com eficiência 4% superior frente ao segundo colocado;
   - Dentre os experimentos entre as técnicas de sumarização, as melhores foram as baseadas em palavras e sentenças, por exemplo, Pontuação de Palavra, Frequência de Termos, além de Posicionamento e Tamanho de Sentenças, utilizando o método de avaliação híbrido proposto.
   - No caso do experimento multilíngue, a plataforma integrada igualou resultados frente aos métodos monolíngues (testados apenas em idioma inglês), além de superar trabalhos relacionados usando o mesmo corpus e configuração ROUGE (TeMário), além de prover os resultados no corpus espanhol, como o corpus foi criado neste trabalho, e por não ser de domínio público, não existem resultados de outros métodos sobre tal corpus.
   - Por fim, além da plataforma de sumarização multilíngue, têm-se as contribuições da criação dos maiores corpora para testes de sumarização já registrados (CNN Inglês e Espanhol), além de ser um trabalho multilíngue que se pôs à prova com mais de dois idiomas presentes em seus experimentos.

1.5 Publicações Geradas ao Longo do Doutorado

Trabalhos completos publicados em periódicos

1. MELLO, R. F.; **CABRAL, L. S.**; FREITAS, F.; LINS, R. D.; SILVA, G. F.; LIMA, R. J.; SIMSKE, S. J.; RISS, M. *A multi-document summarization system based on statistics and linguistic treatment.* Expert Systems with Applications, v. 41, Issue 13, p. 5780–5787, 2014. (Qualis A1)

   Contribuição: Adaptação de algoritmos de pontuação de palavras, sentenças e TextRank para o propósito do artigo.
2. MELLO, R. F.; **CABRAL, L. S.**; LINS, R. D.; SILVA, G. F.; FREITAS, F.; LIMA, R. J.; SIMSKE, S. J.; FAVARO, L. *Assessing Sentence Scoring Techniques for Extractive Text Summarization.* Expert Systems with Applications, v. 41, p. 3082-3094, 2013. (Qualis A1)

   Contribuição: Desenvolvimento de parte das técnicas; aplicação de método híbrido de avaliação; fornecimento do corpus; escrita de parte do trabalho.

Trabalhos completos publicados em anais de congressos

1. MELLO, R. F.; FREITAS, F.; LIMA, R. J.; LINS, R. D.; **CABRAL, L. S.**; SILVA, G. F. *A Four Dimension Graph Model for Automatic Text Summarization.* The IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2013, v. 1, p. 778-781, 2013. (Qualis A1)

   Contribuição: Adaptação de algoritmo TextRank; aplicação de método híbrido de avaliação, fornecimento do corpus utilizado; escrita de parte do trabalho.
2. **CABRAL, L. S.**; LINS, R. D.; LIMA, R. J.; SIMSKE, S. J. *A comparative assessment of language identification approaches in textual documents.* In: IADIS International Conference Applied Computing 2012, 2012, Madrid. Proceedings of IADIS International Conference Applied Computing 2012. Madrid: IADIS, 2012. p. 67-74. (Qualis B2)

   Contribuição: Desenvolvimento dos algoritmos; experimentos e escrita.
3. LINS, R. D.; SIMSKE, S. J.; **CABRAL, L. S.**; SILVA, G. F.; LIMA, R. J.; MELLO, R. F.; FAVARO, L. *A multi-tool scheme for summarizing textual documents.* In: IADIS International Conference WWW/Internet 2012, 2012, Madrid. Proceedings of IADIS International Conference WWW/Internet 2012. Madrid: IADIS, 2012. p. 409-414. (Qualis B2)

   Contribuição: Desenvolvimento da aplicação aplicando o esquema proposto; formação do corpus, escrita de parte do trabalho.

4. MELLO, R. F.; **CABRAL, L. S.**; LINS, R. D.; FREITAS, F.; SIMSKE, S. J.; LIMA, R. J.; FAVARO, L.; SILVA, G. F. *A Context Based Text Summarization System.* Proceedings of 11th IAPR International Workshop on Document Analysis Systems, 2014. (Qualis B1)
   Contribuição: Desenvolvimento de parte dos algoritmos; aplicação de método híbrido de avaliação; fornecimento do corpus utilizado; escrita de parte do trabalho.
5. **CABRAL, L. S.**; LIMA, R. J.; LINS, R. D.; MELLO, R. F.; FREITAS, F.; SILVA, G. F.; SIMSKE, S. J.; FAVARO, L. *A Hybrid Algorithm for Automatic Language Detection on Web and Text Documents.* Booklet of 11th IAPR International Workshop on Document Analysis Systems, 2014. (Qualis B1)
   Contribuição: Desenvolvimento dos algoritmos do trabalho; experimentos, escrita do trabalho.
6. **CABRAL, L. S.**; LINS, R. D.; MELLO, R.; FREITAS, F.; ÁVILA, B.; SIMSKE, S.; RISS, M. *A Platform for Language Independent Summarization.* In: Proceedings of the 14th ACM Symposium on Document Engineering 2014 (DocEng'14) Fort Collins, Colorado, US. September 16-19, 2014. (Qualis B1)
   Contribuição: Desenvolvimento dos algoritmos do trabalho; experimentos, escrita do trabalho.

Em processo de finalização da escrita

1. **CABRAL, L. S.**; ÁVILA, B.; MELLO, R.; LINS, R. D.; FREITAS, F.; SIMSKE, S.; RISS, M. *A New Language Independent Summarization Method.* Meta: Expert Systems with Applications.
   Contribuição: Desenvolvimento dos algoritmos do trabalho; proposta do método; experimentos, escrita do trabalho.

1.6 Estrutura do documento

Este documento adota a utilização de conceitos e tecnologias de PLN para criação de uma plataforma de sumarização multilíngue, abordando-se neste a motivação para o desenvolvimento da solução proposta; no capítulo 2 apresentam-se detalhes sobre identificação automática de idiomas; no capítulo 3, concentram-se as análises e experimentos referentes à sumarização automática de textos; no capítulo 4, trata-se da sumarização extrativa em detalhes; no 5º capítulo tem-se uma descrição detalhada da Sumarização Independente de Idioma, seguida das considerações finais e trabalhos futuros, e por fim relacionam-se as referências e os apêndices usados neste trabalho.

2 Identificação Automática de Idiomas

2

Buscar e aprender, na realidade, não são mais do que recordar.
Platão

Já que você tem que pensar de qualquer forma, pense grande.
Donald Trump

A Identificação Automática de Idiomas em textos é um problema explorado há muito tempo que consiste em determinar em que língua certo texto é escrito. Este problema é amplamente discutido e documentado na literatura, por exemplo, nas revisões (SIBUN & REYNAR, 1996) e HUGHES *et al.*, (2006), apresentam muitos casos, aqui atualiza-se a revisão, apresentando diversas abordagens de sucesso. A área hoje segue auxiliando em ramos de pesquisa modernos como o de reconhecimento de idiomas em textos curtos, e.g. *twitter* (CHANG & LIN, 2014; MOCANU, *et al.*, 2013), análise de sentimento (LIU & ZHANG, 2012), tal evolução incentiva à produção constante de artigos, dissertações e teses atestando a relevância da área, assim como da própria comunidade de processamento de linguagem natural que continua forte e produtiva intelectualmente.

Alguns pesquisadores ressaltam em seus trabalhos que o problema de identificar automaticamente um idioma em textos escritos é menos árduo do que a classificação automática de textos falados, por envolver razões diversas conforme é descrito por (MUTHUSAMY & SPITZ, 1996), onde uma dessas ressalta que o problema de reconhecimento de voz já é trabalhoso por si só, se forem adicionadas tarefas tais como a identificação de idiomas de uma voz reconhecida, tornaria a solução muito mais complexa que no caso da identificação de palavras. Ainda segundo os autores, essa afirmação torna-se intuitiva pelo texto não apresentar a variabilidade associada com a fala, por exemplo: sotaques, emoções e dialetos. Tais fatores contribuem para o aumento dos problemas no reconhecimento de fala e identificação de linguagem falada em detrimento aos problemas de reconhecimento de texto e a identificação da linguagem utilizada.

Por outro lado, o problema revela-se tentador para grande parte dos pesquisadores,

observando-se que o tema foi e é relativamente bem estudado há décadas com as mais diversificadas abordagens, técnicas e resultados. Por consequência, os resultados das pesquisas são amplamente utilizados no mercado, como por exemplo, no contexto dos motores de busca atuais, que há algum tempo investem o poder da Internet aos detentores, identificadores e recuperadores rápidos de informação, tal qual o Google.

Neste capítulo, analisa-se uma coleção de métodos para identificação automática de idiomas em textos escritos e na web, salientando suas vantagens e desvantagens, revisitando pesquisas recentes e algumas já conhecidas e publicadas, inspirando-se em pesquisas similares executadas por (SIBUN & REYNAR, 1996) e HUGHES e colegas (2006). Tendo como fator motivador a criação de um novo método híbrido com base nos pesquisados que seja simples e eficiente, visando adequá-lo à realidade da atividade fim desta tese, antes de efetivamente sumarizar textos em diferentes idiomas, é notória a necessidade da tarefa de identificação a priori dos idiomas dos documentos a serem tratados. Tal procedimento deve ser simples e objetivo para que não utilize recursos computacionais em demasia, economizando-se para as demais fases, como a tradução e a sumarização.

Na próxima seção serão apresentadas as diferentes abordagens, métodos e técnicas já utilizadas na área de identificação automática de idiomas em textos escritos e na web.

2.1. Abordagens, Métodos e Técnicas.

As pesquisas apontam como pioneiro nesta área o trabalho de (GOLD, 1967) que propôs uma solução para o problema usando classes fechadas, a partir de uma lista de idiomas, analisa-se o assunto relatado selecionando um período de uma das orações do texto, identificando o idioma correto. Os experimentos documentados denotam uma simplicidade segundo o autor, por todos os idiomas apresentarem uma representação ortográfica comum, o que facilita tanto na seleção do período quanto na sua identificação.

As soluções ao longo dos anos foram apresentando variações, como a identificando idiomas a partir do preenchimento de formulários guiados por seres humanos, como em (INGLE, 1976) e (NEWMAN, 1987). Já (BEESLEY, 1988) usa fortes fundamentos matemáticos baseados em criptoanálise e probabilidade como um método de identificação de linguagem. Chegando à utilização de técnicas computacionais mais modernas (para época) como coocorrência estatística baseada em *n-gramas* e análise de discriminante linear segundo os trabalhos de (CAVNAR & TRENKLE, 1994) e (SIBUN & SPITZ, 1994) respectivamente, sendo o primeiro voltado para textos e o segundo para imagens.

A utilização de modelos bayesianos por (DUNNING, 1994); aplicação de modelos baseados em vetores de palavras conforme (DANARSHEK, 1995); aparecem como destaque na área. Em (KIKUI & YOKOSUKA-SH, 1996) se aplica o problema de detecção de codificação e determina-se a linguagem de um texto na web envolvendo nove idiomas e onze tipos de codificação. O fato negativo desse método é que não há correspondências entre codificação e linguagem, entretanto como fato positivo tem-se o trabalho com dados cuja fonte é a web. Outro ramo explorado é a classificação linguística a partir de documentos digitalizados (OCR – *Optical Character Recognition*) como o trabalho de LEE e colegas (1998) mostrando resultados confiáveis mesmo sem considerar *strings*.

Iniciando os anos 2000, tem-se registrada uma patente que envolve identificação automática usando informações de palavra e n-gramas (SCHULZE, 2000) o que indica uma sensível tendência das pesquisas do início do século corrente na área. Métodos e técnicas híbridos surgem visando obter resultados eficientes e eficazes, i.e. (POUTSMA, 2001) onde utiliza-se a técnica Monte Carlo para gerar modelos de n-gramas característicos para cada idioma, que então são usados para identificar a linguagem com base no número de ocorrências.

O trabalho (TEYTAUD & JALAM, 2001) que usa o método *Kernel* com *n-gramas* obtidos por IDF (*Inverse Document Frequency*). Na mesma linha LODHI e colegas (2002) propõem um uso especial do *kernel*, que é um produto interno no espaço de características geradas por subsequências de comprimento *k* (*strings*) que ocorrem no texto, com resultados promissores inclusive para diferentes idiomas e para o agrupamento.

O uso de métodos baseados em similaridade para classificar e categorizar, i.e. (ASLAM & FROST, 2003) demonstrou excelente desempenho em classificação multilíngue. Assim como redes neurais escaláveis, como no trabalho (TIAN & SUONTAUSTA, 2003). Outro trabalho que usa a identificação de idiomas e recuperação de contextos usando n-gramas é o (MCNAMEE & MAYFIED, 2004).

O método utilizado por (LINS & GONÇALVES, 2004) considera o uso de derivados gramaticais distribuídos em modelos de classe fechada, tais como preposições, artigos, entre outros. Tal método identifica o idioma a partir de uma árvore de decisão usando combinação de decisões que resultam na correta identificação de forma rápida, necessitando apenas processar parte do documento, um trabalho inovador pelo fato dos derivados gramaticais serem palavras que em comum são descartadas pelos outros sistemas de processamento de linguagem natural (*stopwords*).

O trabalho desenvolvido por (KRUENGKRAI, *et al.*, 2005) revisita o método *kernels string* com pequena quantidade de dados para treinamento. No trabalho de (MARTINS & SILVA, 2005) onde são abordados tópicos de extração de informação da web, podem usar os metadados extraídos caso sejam úteis e válidos, filtrando o que é necessário e limpando *strings* importantes do corpo do texto, escolhendo com algoritmo o melhor n-grama a ser utilizado com o texto, a escolha é feita a partir de similaridade estatística, com bons resultados para um conjunto de 12 idiomas.

Há trabalhos que utilizam estatística e outros modelos fazendo um método híbrido como (AMINE, *et al.*, 2010) que usa um algoritmo *K-Means* e Similaridade. Uma abordagem adaptada de EI (Extração de Informação) é aplicada em trabalhos relativamente mais modernos, com métodos envolvendo SVM (*Support Vector Machines*), *Kernel* e *n-gramas* para identificação de linguagem, como o trabalho de (BHARGAVA & KONDRAK, 2010) que utiliza SVM com bons resultados, principalmente para o idioma chinês.

Observa-se que o problema foi, e ainda é desafiador e contém respaldo científico, dada a tamanha exploração nas pesquisas científicas até os dias atuais, podendo-se confirmar pelas pesquisas realizadas por (SHUYO, 2010), conseguindo gerar perfis de reconhecimento de idiomas automaticamente a partir de *corpora* de treinamento, mescla técnicas como *Machine Learning*, *K-means* e análise probabilística.

Na próxima seção são apresentados os recursos e tecnologias utilizados e disponíveis para desenvolver um projeto na área de identificação automática de linguagem em textos.

2.2. Recursos e Tecnologias

Existe uma gama de recursos e tecnologias para identificação da linguagem em textos, alguns bem conhecidos e citados na maioria dos trabalhos referenciados na seção anterior, entretanto há ferramentas muito importantes que os pesquisadores ainda não atentaram que podem ser utilizadas em pesquisas tanto direcionadas ao tema quanto híbridas.

Dentre os recursos mais conhecidos e disseminados nos trabalhos da área, tem-se o *TextCat*[1], uma implementação do algoritmo de categorização de texto proposto por (CAVNAR & TRENKLE, 1994) baseado em n-gramas. Outro exemplo é o *Rosette*[2] da *BasisTech* um identificador de linguagem e analisador de texto multilíngue (reconhece 55 idiomas e 45 codificações), identificando a linguagem e o esquema de codificação de

[1] http://odur.let.rug.nl/~vannoord/TextCat/
[2] http://www.basistech.com/language-identifier/

caracteres de forma rápida e precisa. Outro enfoque bem conhecido é o *webservice* da *Xerox* denominado *Language Identifier*[3] que fornece identificação de idioma que frequentemente é o primeiro passo necessário em toda uma linha de processamento de documentos para tradução, podendo-se usar um formulário baseado na web ou uma API (*Application Programming Interface*).

Seguindo o segmento API, tem-se uma ferramenta denominada *language-detection*[4]desenvolvida por (SHUYO, 2010), trata-se de uma biblioteca para detecção de idiomas para a linguagem de programação Java sob licença aberta *Apache 2.0,* usa Aprendizagem de máquina, probabilidade e filtros *naive*-Bayesianos para detectar idiomas obtendo 99% de precisão em média para 53 línguas.

Tem-se ainda o foco comercial, a indústria tem explorado a área também como, por exemplo, a *Alchemy API*[5] uma biblioteca completa para exploração textual, incluindo recuperação, extração, categorização, detecção de idiomas. Finalizando com o exemplo da *Google Translate API*[6] uma ferramenta com custo calculado pela quantidade de dados traduzidos ou identificados, usa métodos que utilizam como entrada uma referência ou documento HTTP retornando o idioma encontrado no documento.

Para uma aplicação mais robusta e que também envolva diversas áreas de conhecimento a arquitetura GATE (*General Architecture for Text Engineering*) (CUNNINGHAM, *et al.*, 2011) é a mais indicada por sua modularidade, que inclui diversos *plugins* que podem ser adicionados à aplicação, por exemplo o *LingPipe* (com vários recursos, inclusive identificador de idiomas), *TextCat* (identificação de idiomas baseada no *TextCat*), *Keyphrase Extraction Module* (módulo com recursos de extração de informação), *Ontology Integration* (módulo de integração com ontologias). Voltada para engenharia textual, permite o processamento de textos em diversos idiomas (53 ao todo) e vem sendo utilizado por grupos de pesquisa da área em todo o mundo, sua principal desvantagem é o alto tempo de processamento utilizado por seus recursos.

Na próxima seção são reportados alguns trabalhos relacionados na área de classificação de idiomas, seus experimentos e uma análise dos resultados divulgados.

[3] http://open.xerox.com/Services/LanguageIdentifier/
[4] http://code.google.com/p/language-detection/
[5] http://www.alchemyapi.com/api/
[6] http://code.google.com/intl/pt-BR/apis/language/translate/v2/getting_started.html

2.3. Trabalhos Relacionados

As pesquisas revisitadas utilizam, em sua maioria, a identificação de idioma em textos escritos, falados ou digitalizados, porém sempre focam um número restrito de linguagens a serem identificadas, o que restringe o tamanho do domínio a ser estudado, diante dos *corpora* utilizados, a precisão é relativa às linguagens mais presentes nos *corpora*, i.e. inglês, em detrimento a linguagens que não estão supridas de uma quantidade de documentos estatisticamente relevante, como por exemplo árabe, muito embora esta afirmação esteja mais caracterizada para uma hipótese, pois não há registro explícito dos *corpora*.

Outro fator interessante são os trabalhos envolvendo classe fechada, tendo uma classificação a partir de uma lista pré-determinada de termos candidatos, que por consequência denota trabalhos com bom desempenho e resultados, entretanto inovador seria o trabalho que utilizasse uma identificação a partir de uma classe aberta, denotando até classificação de linguagens desconhecidas, a partir é claro de algum algoritmo de aprendizado.

Alguns dos trabalhos que envolvem aprendizado ou redes neurais trabalham com um conjunto pequeno de documentos/palavras (cerca de 10%) como *corpus* de treino, caso este número pudesse ser aumentado para 20 ou 30% do total de documentos/palavras, provavelmente o percentual de acerto final seria melhor, diminuindo a incidência de falsos negativos.

Tratando de aprendizado, todos os estudos analisados classificam um documento como sendo de um idioma, entretanto, caso tenha-se um documento multilíngue, que seja escrito 90% em português, 5% em inglês, 3% em espanhol e 2% em francês, provavelmente todos o classificariam como um documento de língua portuguesa, entretanto não seria totalmente correta a classificação, sugere-se que haja uma classificação estatística do documento demonstrando, caso este seja multilíngue, o percentual de cada idioma encontrado no processamento.

Note-se ainda que efeitos de pré-processamento nos *corpora* ajudam a aumentar o resultado final dos trabalhos, indiscutivelmente os recursos de PLN (processamento de linguagem natural) como marcação das partes da fala (*POS Tag*), conhecimento linguístico do documento, dentre outros, resultam na transformação de um documento antes desconhecido, agora conhecido e com metadados, o que pode garantir uma vantagem em relação aos demais.

Deixando de lado o aspecto qualitativo, tem-se uma abordagem quantitativa na comparação dos resultados das técnicas e algoritmos pesquisados. Dentre os mais de 26 trabalhos foram preferidos os que mencionavam em suas pesquisas a acuidade do algoritmo, número de documentos do *corpus*, número de idiomas reconhecidos e tempo de processamento, entretanto muitos não utilizaram tais variáveis e dentre os que utilizaram, poucos as informaram completamente, assim em alguns gráficos e tabelas encontrar-se-ão a sigla N.D. que denota "não divulgado".

Deste modo, a extração dos resultados de cada trabalho foi consolidada na Tabela 1, como muitos dos trabalhos apresentam resultados variados por idioma e método, neste estudo, obteve-se a média da acurácia do método independente aos idiomas aplicados, para não viciar resultados, uma vez que o algoritmo deve funcionar para todos os idiomas a que faz luz em suas publicações.

Tabela 1. Relação entre *Corpus* x Idiomas x Acuidade x Tempo de processamento.

Algoritmo	***Corpus***	**Idiomas**	**Acuidade**	**Tempo**
1) (LINS & GONÇALVES, 2004)(*web*)	492	6	90,92%	0,0681
2) (LINS & GONÇALVES, 2004)(*text*)	100	6	97,50%	0,0671
3) (BEESLEY, 1988)	1	3	100,00%	N.D.
4) (SIBUN & REYNAR, 1996)	892	27	91,43%	N.D.
5) (SIBUN & REYNAR, 1996)	892	3	97,26%	N.D.
6) (Hakinnen e Tian, 02)	N.D.	4	81,16%	N.D.
7) (BHARGAVA & KONDRAK, 2010)(*LM*)	N.D.	4	94,26%	N.D.
8) (BHARGAVA & KONDRAK, 2010)(*LSVM*)	N.D.	3	97,13%	N.D.
9) (MARTINS & SILVA, 2005)	258772	12	87,00%	N.D.
10) AMINE *et al.* (2010)	1141	3	45,85%	116,8333
11) (TEYTAUD & JALAM, 2001)	N.D.	10	75,19%	N.D.
12) (SIBUN & SPITZ, 1994)	N.D.	23	92,65%	N.D.
13) KRUENGKRAI *et al.* (2005)	N.D.	17	95,25%	N.D.
14) LEE *et al.* (1998)	188	5	95,44%	N.D.
15) (CAVNAR & TRENKLE, 1994)	3478	14	97,96%	N.D.
16) (SCHULZE, 2000)	N.D.	9	91,33%	N.D.

Conforme se pode observar, a melhor acurácia é dada pelo algoritmo de (BEESLEY, 1988), entretanto ele utilizou apenas 2 documentos em seu *corpus* que continha somente uma frase cada, um francês e outro espanhol, e efetuava uma relação probabilística entre as palavras para escolha do idioma, o que pode ser afetado caso utilize um documento com 3 frases com mesmo número de palavras em 3 idiomas diferentes, a relação probabilística vai atribuir a um dos idiomas, o que não corresponde à realidade, um documento multilíngue.

Dentre os algoritmos estudados, os que mesclam uma boa precisão, quantidade de documentos relevante estatisticamente no *corpus* e boa quantidade de idiomas reconhecidos são (LINS & GONÇALVES, 2004), (SIBUN & REYNAR, 1996), (BHARGAVA & KONDRAK, 2010), (SIBUN & SPITZ, 1994), KRUENGKRAI *et al.* (2005), LEE *et al.* (1998), (CAVNAR & TRENKLE, 1994) e (SCHULZE, 2000) conforme pode-se observar na Tabela 1 e na Figura 4.

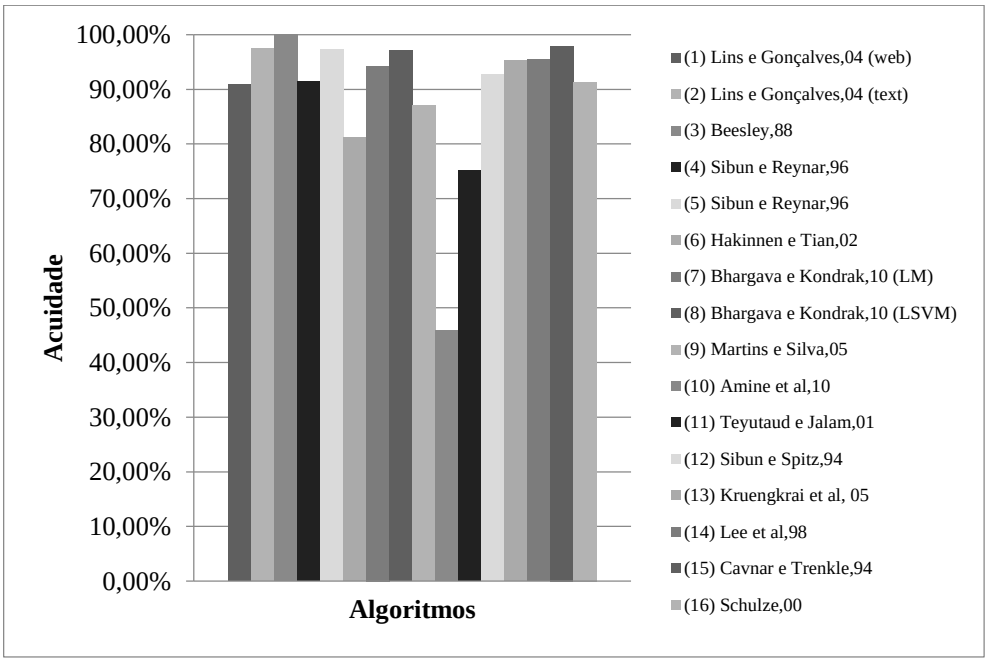

Figura 4. Acuidade média.

Em destaque, dentre os trabalhos preteridos enaltece-se o trabalho de (MARTINS & SILVA, 2005) embora tenha precisão média de 87% utilizaram um *corpus* grande com aproximadamente uma média de 260 mil documentos web por idioma (a maioria português), pouco mais de três milhões no total. Com relação ao número de idiomas reconhecidos, destaque para (SIBUN & REYNAR, 1996), (MARTINS & SILVA, 2005), (TEYTAUD & JALAM, 2001), (SIBUN & SPITZ, 1994), KRUENGKRAI *et al.* (2005) e (CAVNAR & TRENKLE, 1994) todos reconhecendo mais de 10 idiomas com boa precisão e *corpus* com relevância estatística.

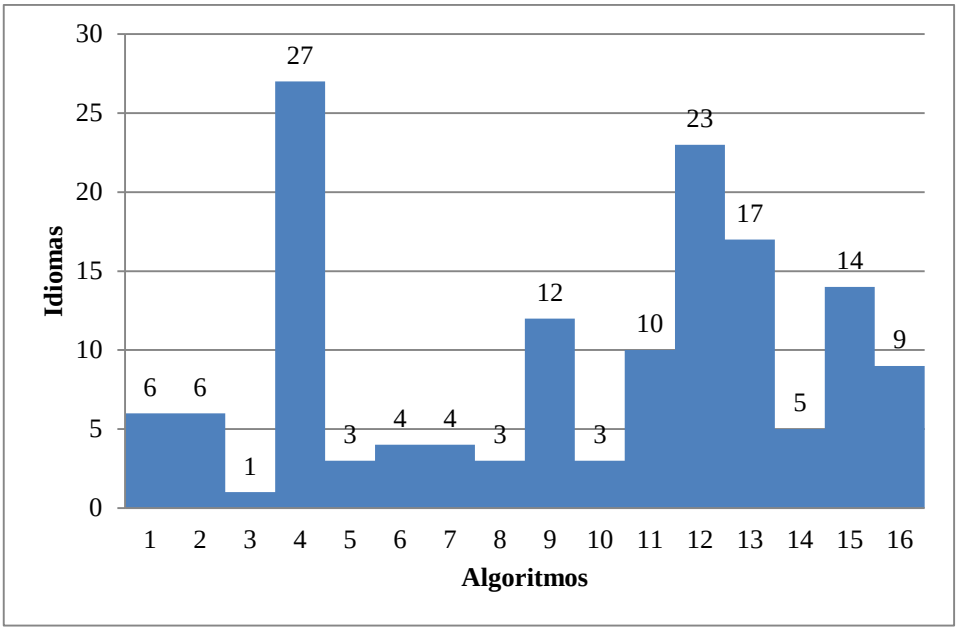

Figura 5. Número de idiomas reconhecidos.

Os trabalhos com maior média de documentos por idioma foram o de (MARTINS & SILVA, 2005) (conforme mencionado nos parágrafos anteriores), o de (CAVNAR & TRENKLE, 1994), AMINE e colegas (2010), (SIBUN & REYNAR, 1996) e (LINS & GONÇALVES, 2004) denotam os trabalhos com *corpus* mais relevantes em termos estatísticos, trabalhando com no mínimo 490 documentos por idioma em média.

Uma característica comum a quase todos os trabalhos é a não divulgação do tempo de processamento do algoritmo, em sua grande maioria omitem tal informação, dois autores ainda informam a quantidade de memória utilizada no momento, porém apenas (LINS & GONÇALVES, 2004) e AMINE e colegas (2010) divulgaram os tempos de processamento dos seus algoritmos.

Diante do exposto pode-se afirmar que os resultados dos trabalhos são interessantes, contudo, abstendo a concepção e esforço empregado nas pesquisas, mensurando quantitativamente, destacam-se os trabalhos de (LINS & GONÇALVES, 2004) pela boa precisão tanto em *corpus* textuais e web, com algoritmo simples e tempo de processamento excelente. Já SIBUN & REYNAR (1996), MARTINS & SILVA (2005), CAVNAR & TRENKLE (1994) e LEE *et al.* (1998) embora não tenham divulgado suas medidas temporais, obtiveram bons resultados com *corpora* de no mínimo, 100 itens. Por fim (SIBUN & SPITZ, 1994), (SCHULZE, 2000), KRUENGKRAI *et al.* (2005), (BHARGAVA & KONDRAK, 2010) e (BEESLEY, 1988), que por não divulgarem a quantidade de documentos utilizados ou ter utilizado quantidade muito pequena, obtiveram resultados

expressivos em seus trabalhos, o que não retira o mérito dos cientistas em suas pesquisas.

Tais méritos que são reconhecidos até hoje pela comunidade científica, o que denota o tópico de classificação de idiomas como frequente em documentos publicados em recentes conferências, como por exemplo, (BOTHA & BARNARD, 2012) que analisam os fatores que afetam a precisão da classificação de idiomas em documentos de texto; já (KAMPER & NIESLER, 2012) apresentam uma revisão literária sobre identificação de dialetos, idiomas e sotaques; GEIGER e colaboradores (2012) apresentam um enfoque com n-gramas de tamanho reduzido. Assim como o trabalho de CABRAL e colegas (2012) que denotam experimentos interessantes com algoritmos de bom desempenho usando corpus com sentenças reduzidas, e mais adiante CABRAL e colegas (2014) reforçam a tese com um corpus mais encorpado.

Observando-se o contexto geral, nota-se que cada trabalho é desenvolvido em um ambiente de hardware e software diferente, utilizando recursos e corpora variados, conclui-se deste modo, que para uma análise e avaliação cientifica independente e justa, seria necessário montar um ambiente de experimentos homogêneo, incluindo recursos e corpora. Assim propõe-se a partir da próxima seção, um novo algoritmo para identificação de idiomas em textos, além disso, propõe-se um ambiente de testes homogêneo, com o mesmo hardware, software e corpora para validar a proposta e seus concorrentes, espera-se que a solução proposta atinja resultados iguais ou superiores aos concorrentes, primando pelo tempo de processamento, visando economia de recursos para a tarefa principal, a sumarização.

2.4. Solução Proposta

Antes de se iniciar algum experimento ou avaliação, foi necessário implementar um conjunto de algoritmos sob uma mesma linguagem de programação (no caso, Java), para fins de isenção científica e validação. As descrições dos métodos para identificação de idioma são abordadas nesta seção, tais métodos foram escolhidos diante da clareza de definição de seus processos, sendo possível sua replicação.

Após a revisão da literatura, codificaram-se três algoritmos baseados em propostas da literatura, além de um novo algoritmo, a saber: (i) CALIG, (ii) CALIM, (iii) TextCat e (iv) LangDetect. Em particular, deve-se descrever e analisar em detalhes a solução proposta, o método para classificação de idiomas batizado como CALIM, apresentando seus pormenores subjacentes, bem como uma explicação detalhada do seu funcionamento.

O Algoritmo CALIG

O algoritmo apresentado em (LINS & GONÇALVES, 2004) faz uso de classes gramaticais fechadas de cada idioma como dicionário visando uma forma rápida de identificar o idioma de um documento. Utiliza-se uma árvore de decisão com condições organizadas em cascata, assim em um primeiro passo, conta-se o número de advérbios em inglês, em um segundo passo conta-se o número de preposições em espanhol, ou seja, contam-se as classes mais significativas para cada idioma. Tais priorizações de classe fechada foram decididas a priori através da análise estatística dos vários documentos.

No trabalho original de Lins e Gonçalves abrange seis idiomas, por isso, neste trabalho incrementou-se este número estendendo-se seguindo o método originalmente descrito em (LINS & GONÇALVES, 2004) com objetivo de cobrir todas as 21 línguas utilizadas pelo corpus *Europarl* (KOEHN, 2005), que foi escolhido inicialmente como padrão para os experimentos. *Insuma*, o algoritmo funciona como um reconhecedor, usando como padrão, dicionários gramaticais priorizando algumas classes fechadas de cada idioma, e.g. advérbios, preposições, numerais, artigos, pronomes, conjunções e interjeições.

O idioma do texto é determinado após a leitura do documento de entrada e então a classificação através da árvore de decisão. Se pelo menos 5 palavras são encontradas nos dicionários, desde que o percentual de palavras do idioma candidato seja no mínimo 40% superior ao do segundo dicionário, pode-se dizer que o documento foi escrito no idioma associado ao dicionário mais bem classificado. Tais critérios de decisão foram alterados nesta versão, em virtude do *corpus* escolhido conter uma variação grande no tamanho das sentenças, abrangendo sentenças de apenas 3 *tokens*, assim como outras maiores.

Deste modo, necessitou-se incluir uma camada léxica que testará o idioma de acordo com suas características de escrita e acentuação, tendo-se registrado máximo de itens léxicos candidatos, este idioma leva vantagem frente aos demais de imediato. Os padrões léxicos são singulares, tais como “ão” que são especialmente da língua Portuguesa, “ß” para o alemão, assim como o “æ” para o dinamarquês. A inclusão de tal padrão de detecção aumenta, no geral, a capacidade de classificação e precisão do algoritmo, além de diminuir o tempo de processamento.

O método CALIM

O método de CALIM de identificação de idiomas foi inspirado nas ideias de

(DUNNING, 1994), (LINS & GONÇALVES, 2004) e (CAVNAR & TRENKLE, 1994). O algoritmo CALIM é baseado em dicionários que são perfis idiomáticos, que leva em consideração a frequência relativa de janelas de caracteres (*n-gramas*) presentes em todas as línguas em estudo (21 no total) descrito em CABRAL e colegas (2012). Mais precisamente, a criação de tais perfis leva-se em conta aproximadamente 250 *n-gramas* mais frequentes para cada idioma. Para a criação desses perfis idiomáticos, utilizam-se alguns dados fornecidos pelo dicionário *Lexiteria* que é uma iniciativa que visa à compreensão de vários aspectos da linguagem humana (LEXITERIA, 2012). *Lexiteria* é uma empresa especializada em fornecer listas de palavras, incluindo listas de palavras frequentes, n-gramas, bem como glossários e grupos de dicionários personalizados.

Diante dos propósitos de investigação, recolheram-se alguns dados estatísticos para 21 idiomas, tais como frequência de palavras, comprimento médio de palavra, entre outros. Após criar os dicionários de cada idioma, ordenados pelas palavras mais frequentes em ordem decrescente selecionaram-se as primeiras *M* palavras, dadas pela fórmula:

$$M = \left(\frac{\sum_{i=1}^{n} length(w_i)}{n} \right)^2 \quad (1)$$

A hipótese seguida pelo algoritmo CALIM leva em consideração que será mais provável encontrar palavras de alta frequência no texto de entrada em relação às de baixa frequência. Além disso, por motivo de desempenho, apenas consideram-se as palavras de comprimento máximo 5 ($n \leq 5$). Justifica-se essa escolha porque, na maioria dos idiomas, as palavras de maior frequência são preposições, numerais, pronomes pessoais, entre outras, são caracteristicamente menores. Deste modo, caso uma palavra contenha mais do que 5 caracteres, tomar-se-á como *n-grama* seu sufixo de tamanho=5 para ser incluído no perfil idiomático. Os experimentos levaram em consideração outros valores de *n*, mas os melhores resultados entre acurácia e desempenho foram obtidos utilizando $n = 5$.

Durante a etapa de classificação do algoritmo CALIM, aplica-se uma heurística muito útil que contribuiu para resultados mais precisos em nossos experimentos. Esta heurística assume que, se uma palavra (ou *token*) é composto n-gramas muito específicos encontrados exclusivamente em determinados idiomas (i.e. "*ão*" no idioma Português, "*ñ*" no Espanhol, etc.), então o método atribui um valor superior ao que é atribuído no esquema de classificação normal que é 1.

Adotou-se o critério que o idioma mais bem classificado será o escolhido. No final da etapa preliminar de classificação, cada *token* do texto de entrada é classificado em 1 ou

mais idiomas. No caso de dois ou mais idiomas ao final terminarem empatados, procede-se com um passo adicional, que consiste em multiplicar a pontuação final de cada *token* pela frequência relativa normalizada de cada um. A frequência normalizada de um *token* é simplesmente calculada pela razão entre a frequência e a soma aproximada de todas as frequências dos *tokens* para cada perfil idiomático. Esta simples heurística contribuiu para escolher o idioma correto do documento.

TextCat

Existem diversas ferramentas e plataformas para classificação de idioma em textos, algumas muito conhecidas e referenciadas pela maioria dos trabalhos listados na última seção, entre os mais conhecidos e utilizados recursos, encontra-se o *Java Text Categorizing* (TextCat)[7] *library* (Knallgrau New Media Solutions, 2012). Ela consiste de uma implementação em Java puro da biblioteca LibTextCat[8] para classificação de idiomas escrita originalmente em linguagem de programação C.

TextCat é distribuído sob a licença LGPL[9] e também pode ser usado para categorizar textos em tópicos arbitrários pela computação adequada de características individuais que representam as categorias, como foi inicialmente proposto em (CAVNAR & TRENKLE, 1994). Este algoritmo adota um modelo de *n-gramas* para representar um documento, que parece ser a abordagem mais promissora. A ideia central deste algoritmo é calcular uma característica individual de um documento associado a uma categoria desconhecida, e compará-la com as características de certo número de documentos nos quais as categorias são conhecidas. A característica individual é obtida por uma lista dos *n-gramas* mais frequentes que ocorrem em um documento, ordenado pela frequência.

Estas características individuais (chamadas pelos autores de *fingerprints*) são comparadas utilizando uma medida chamada *out-of-place*, que se assemelha à medida da entropia (CAVNAR & TRENKLE, 1994). Por fim, as categorias das correspondências mais próximas são a saída para a classificação. A principal vantagem deste algoritmo é o suporte para identificação de idioma de textos ruidosos, por exemplo, textos provenientes de sistemas OCR. Entre as suas desvantagens, é importante ressaltar que a formação de categorias requer tempo e falta-lhe suporte para idiomas importantes como o português.

[7] http://textcat.sourceforge.net/
[8] http://software.wise-guys.nl/libtextcat/
[9] GNU Lesser General Public License, http://www.gnu.org/copyleft/lesser.html

Language Detector (LangDetect)

É composto de uma biblioteca para classificação de idiomas desenvolvida em Java sob *Apache Open License*. O método foi implementado por (SHUYO, 2010) com base em técnicas propostas por (DUNNING, 1994). Em sua abordagem, Dunning assume que o idioma pode ser modelado pelo processo de Markov de baixa ordem que gerando *tokens*, e, em seguida, usando regras de decisão Bayesiana para classificá-los. Além disso, o autor também afirma ter 99% de precisão em média na discriminação entre dois idiomas moderadamente relacionados, como o inglês e o espanhol.

Dicionário (*SimpleDic*)

É um método que utiliza dicionários simples para cada idioma, desenvolvido utilizando como principal componente, *stopwords* comuns do Processamento de Linguagem Natural (PLN), que por padrão, são ignorados na maioria das aplicações PLN, aqui se encontrou uma utilização funcional, diante da frequência em que estes termos ocorrem no texto, do tamanho e da invariabilidade, em sua maioria.

2.5. Configurações dos Experimentos

A avaliação dos desempenhos dos quatro métodos codificados para classificação automática de idiomas é efetuada nesta seção. Após apresentar a configuração adotada nos experimentos, a aferição dos métodos, discutir-se-ão os resultados alcançados.

2.5.1. Ambiente de Testes

Todo o trabalho de pesquisa envolvendo classificação de idioma em textos se concentra em um número limitado de idiomas, o que restringe o tamanho do domínio a ser estudado por causa dos corpora utilizados. Como poucos disponibilizam seus recursos para outros pesquisadores, torna-se difícil confirmar a validade dos resultados registrados. Nesse contexto, esta seção analisa quatro algoritmos de classificação de idiomas distintos de forma justa e uniforme. Todos os algoritmos foram codificados na mesma linguagem de programação e foram testados sobre o mesmo *corpus*.

Dentre os vários *corpora* disponíveis, o *European Parliament Proceedings Parallel Corpus* ou simplesmente *EuroParl Corpus* v7 (KOEHN, 2005) merece uma atenção especial para os nossos fins de investigação. O corpus é composto de documentos relatando discursos e debates ocorridos no Parlamento Europeu desde 1996. O corpus compreende documentos em 21 idiomas presentes na Comunidade Europeia e que foi usado em competições

patrocinadas por associações e conferências internacionalmente conhecidas, como a Associação de linguística computacional (ACL), a Conferência sobre métodos empíricos em processamento da linguagem Natural (EMNLP), e a oficina de tradução automática (WMT), apenas para citar algumas.

Para a análise comparativa aqui descrita, utiliza-se o *Europarl Corpus*, que é composto por duas versões originalmente, a "*test*" que contém 21.000 documentos coletados no período 1996-2011, divididos igualmente entre 21 idiomas europeus, com número pequeno de sentenças por documento em formato textual sem formatação; e outra versão é denominada "*full*" na qual contém cerca de 60.000 documentos em formato XML, com distribuição variável da quantidade de documentos por idioma suportado (búlgaro, checo, dinamarquês, alemão, inglês, estoniano, finlandês, francês, holandês, grego, húngaro, italiano, letão, lituano, polonês, português, romeno, eslovaco, esloveno, espanhol e sueco). As versões serão utilizadas em experimentos separados. Além disso, foi registrado o tempo decorrido (em segundos) na execução de cada experimento. Essas medidas foram tomadas através de um computador equipado com processador Intel Core i3-2330M 2.20 GHz, 4 GB RAM, executando o sistema operacional Windows (versão 7 - 64 bits).

2.5.2. Resultados e Discussão

De todos os experimentos relatados na presente seção, leva-se em conta a equidade da comparação. A Tabela 2 fornece a média da precisão dos resultados obtidos na primeira rodada de testes usando todos os algoritmos avaliados em dois subconjuntos do *Europarl Test Corpus* composto de 5 e 13 idiomas.

Em primeiro lugar tem-se que considerar a extensão do algoritmo CALIG para identificar mais 15 idiomas, ou seja, 21 idiomas encontrados no *Europarl Corpus* relacionado aos seis idiomas inicialmente abrangidos pelo algoritmo (LINS & GONÇALVES, 2004). Como já relatado anteriormente, o algoritmo é baseado em classes gramaticais fechadas, onde se teve dificuldade em encontrar esse tipo de dicionários para complementar os idiomas. Como uma possível solução para o problema, utiliza-se o *Wiktionary*[10], um projeto colaborativo para produzir conteúdo gratuito, no caso, dicionário multilíngue, serviu como uma fonte para formação da lista de classes gramaticais fechadas.

Conforme a Tabela 2, embora o método CALIG apresente bons resultados para cinco idiomas, sua precisão diminui com 13 idiomas. Um dos possíveis motivos para explicar esse

[10] http://en.wiktionary.org/wiki

tipo de comportamento, como já mencionado no capítulo 3, é que este método foi originalmente desenvolvido para apenas seis idiomas, considerando que o CALIM e LangDetect pode identificar uma quantidade maior de idiomas.

Um aspecto positivo do CALIG é que ele apresenta o segundo menor tempo de processamento entre todos os métodos selecionados. Isso é explicado pelo fato de que o método tem um dicionário menor em comparação com outros métodos, o que significa menos comparações efetuadas pelo classificador. Estes resultados sugerem que, com um processo de aprendizagem de máquina pode-se estender os dicionários suportar uma maior quantidade de idiomas, embora o método possa ser uma boa opção para casos em que o tempo de resposta seja a variável de maior importância.

Os outros métodos não apresentaram diferença estatisticamente significante entre si no que diz respeito à precisão dos resultados com cinco idiomas, mas o tempo de processamento varia muito. Nos experimentos efetuados, a execução do TextCat tomou muito mais tempo conforme pode ser visto nas Tabelas 2 e 3.

Tabela 2. Os resultados médios com 5 e 13 idiomas (Europarl *Test* Corpus)

Métodos	**5 idiomas**[11]		**13 idiomas**[12]	
	Precisão	**Tempo(s)**	**Precisão**	**Tempo(s)**
CALIG	93,68	3,78	64,74	9,83
TextCat	97,10	137,87	88,37	358,47
CALIM	99,02	3,01	96,36	7,08
LangDetect	99,48	3,82	98,92	9,95

Pode-se observar que o TextCat ficou aquém dos demais em relação ao tempo de processamento, além de não suportar todos os 21 idiomas contidos no corpus, assim para evitar comparações equivocadas, decidiu-se retirar o método dos experimentos a serem executados a partir deste ponto.

Tabela 3. Os resultados médios com 6 e 21 idiomas (Europarl *Test* Corpus)

Métodos	**6 idiomas**		**21 idiomas**	
	Precisão	**Tempo(s)**	**Precisão**	**Tempo(s)**
CALIG	94,72	4,54	-	-
CALIM	98,93	3,60	97,08	12,62
LangDetect	99,45	4,59	99,16	16,08

[11] Alemão, inglês, espanhol, francês e italiano.
[12] Dinamarquês, alemão, inglês, espanhol, finlandês, francês, húngaro, italiano, holandês, polonês, eslovaco, esloveno e sueco.

Nos resultados experimentais apresentados na Tabela 3, conforme já registrado, não se considera o TextCat devido ao limitado número de idiomas suportados e seu alto tempo de processamento. Assim, os resultados experimentais com os demais métodos com seis idiomas, são bastante semelhantes aos apresentados na Tabela 2. Analogamente, não são apresentados os resultados obtidos pelo CALIG com 21 idiomas devido os seus dicionários não estarem completamente revisados para idiomas como estoniano, polonês e eslovaco.

Os algoritmos CALIM e LangDetect obtiveram alta precisão em todos os experimentos desta primeira rodada. O LangDetect obteve mais de 99% de precisão (conforme prometido em seu trabalho) e requereu apenas 16 segundos para processar 21.000 documentos de texto. O algoritmo mais rápido foi o CALIM, embora tenha obtido uma precisão de 0,52 ponto percentual abaixo do LangDetect, precisou de apenas 12 segundos para processar a mesma quantidade de documentos.

Nesta primeira fase de experimentos, pode-se considerar LangDetect como algoritmo mais preciso, tendo-se em segundo o CALIM, proposto neste trabalho, contudo considerando-se o desempenho combinado entre acurácia e tempo de processamento, pode-se afirmar que o algoritmo CALIM apresenta melhor custo benefício.

Um fator que pode dificultar a identificação precisão está relacionado ao tamanho do texto de entrada, que ainda é um desafio para classificadores de idiomas. Para o corpus selecionado, há sentenças muito curtas detectados, abrangendo apenas 4 *tokens*, i.e. "*Mødet åbnet kl. 09.00*" para dinamarquês.

Assim, visando a avaliar a sensibilidade dos algoritmos selecionados em função de uma amostra variando o comprimento (em caracteres) do texto de entrada, foram realizados dois outros experimentos. As Tabelas 4 e 5 apresentam tanto a precisão e tempo de CPU obtidos para amostras textuais em 13 e 21 idiomas, respectivamente.

Tabela 4. Análise de sensibilidade ao *Europarl Test Corpus* precisão e tempo de processamento (13 idiomas)

Tamanho (caracteres)	**Precisão**			**Tempo de CPU**		
	TextCat	**LangDetect**	**CALIM**	**TextCat**	**LangDetect**	**CALIM**
10	0,54	0,65	0,47	42,78	11,97	4,62
20	0,69	0,85	0,69	73,44	11,59	4,81
30	0,76	0,93	0,80	107,96	11,60	5,18
40	0,80	0,96	0,86	132,83	11,50	5,26
50	0,82	0,98	0,90	175,75	11,49	5,67
75	0,84	0,99	0,94	242,10	11,99	6,55
100	0,85	0,99	0,96	272,92	11,22	6,59

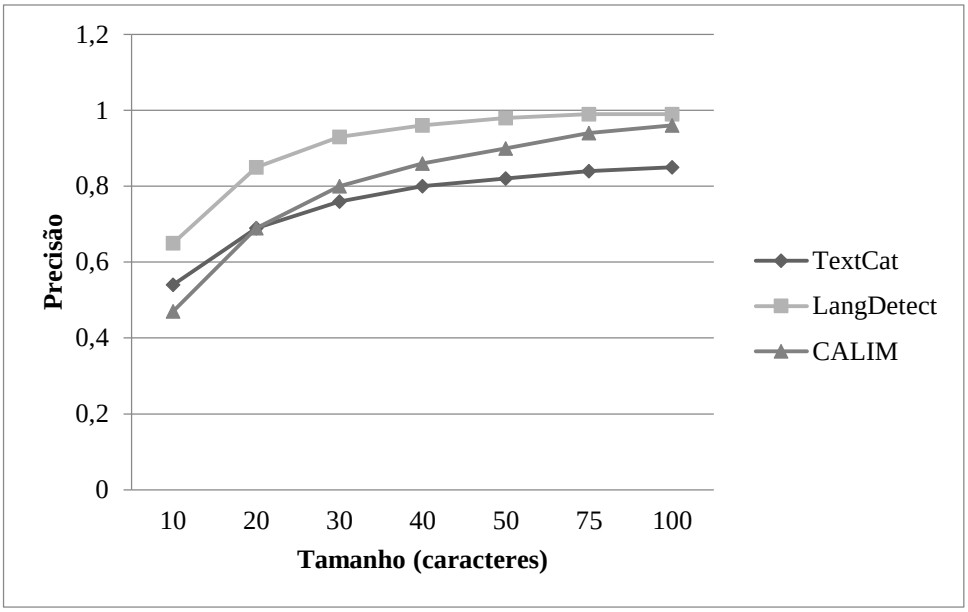

Figura 6. Gráfico de sensibilidade ao *Europarl Test Corpus* precisão (13 idiomas)

Outra vez, o TextCat comportou-se bem, mas seu tempo de processamento mostra-se aquém dos demais, deste modo no experimento para 21 idiomas ele foi retirado, também em virtude da falta de suporte para alguns idiomas presentes no corpus.

Tabela 5. Análise de sensibilidade ao *Europarl Test Corpus* precisão e tempo de processamento (21 idiomas)

Tamanho (caracteres)	**Precisão**		**Tempo de CPU**	
	LangDetect	**CALIM**	**LangDetect**	**CALIM**
10	0,68	0,53	15,08	6,92
20	0,86	0,74	15,94	7,72
30	0,93	0,84	15,40	7,88
40	0,96	0,89	15,06	8,39
50	0,97	0,92	15,52	8,78
75	0,98	0,96	15,45	9,58
100	0,99	0,97	15,41	10,36

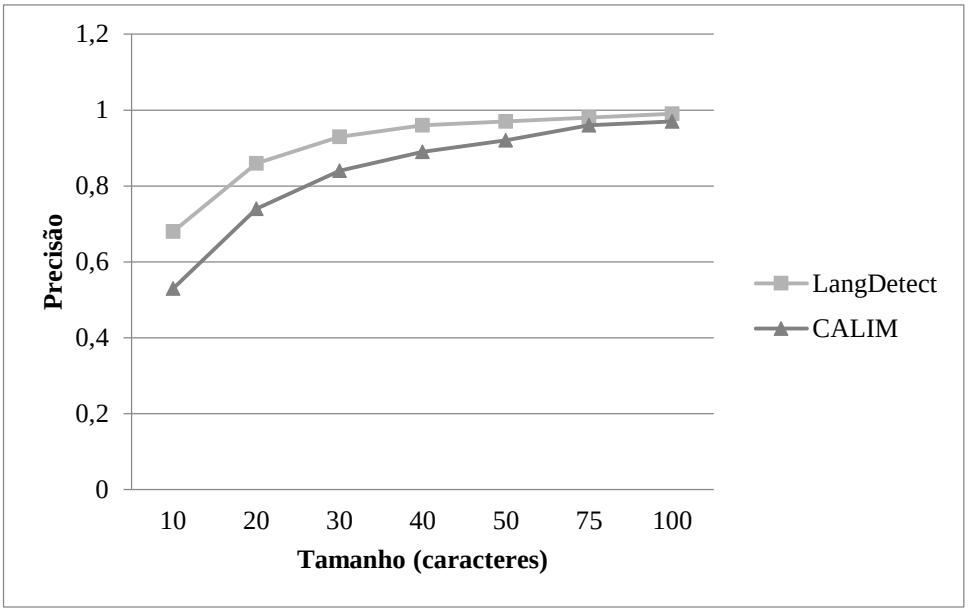

Figura 7. Gráfico de sensibilidade ao *Europarl Test Corpus* precisão (21 idiomas)

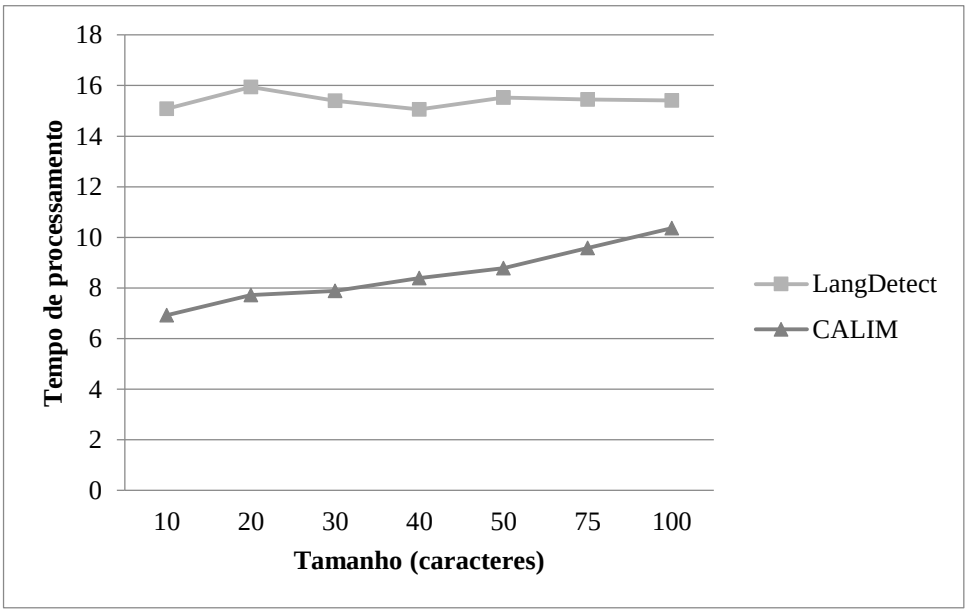

Figura 8. Gráfico de sensibilidade ao *Europarl Test Corpus* tempo de processamento (21 idiomas)

A análise das Tabelas 4 e 5 e Figuras 7 e 8 revela-nos que no TextCat a formação da estrutura definida no algoritmo para classificar o idioma toma um tempo demasiadamente grande em relação aos demais na sua implementação em Java. Com apenas 50 caracteres de entrada, o LangDetect consegue classificar com precisão de 98% para 13 idiomas, e de 97% para 21 idiomas. Por outro lado, o CALIM, para 13 e 21 idiomas, obteve precisão de 90% e 92% respectivamente, já no tempo de processamento, é notório que o CALIM é quase 50% mais rápido que o LangDetect, devido a sua simples formação de perfis idiomáticos além de

solução algorítmica que utiliza aritmética simples para calcular os pesos e decidir sobre o idioma correto. Isso serve como prova de que ambos os métodos não necessitam de textos extensos como entrada para obtenção de bons resultados.

Em uma terceira e última rodada de experimentos, após uma revisão nos perfis e dicionários utilizados em alguns dos algoritmos, além de incluir uma fase de pré-processamento do corpus, para que se aceitem tanto, documentos de texto sem formatação, quanto, documentos de texto em formato XML.

Diante disso efetuaram-se os experimentos com o corpus na versão *Test* e também na versão *Full*, obtendo os seguintes resultados. Ao repetir os experimentos com a versão *Test* obteve-se resultados semelhantes aos anteriormente obtidos, retirando o TextCat e adicionando o método de dicionário simples.

Tabela 6. Os resultados médios com 6 e 21 idiomas (Europarl *Test* Corpus)

Métodos	**6 idiomas[13]**		**21 idiomas[14]**	
	Precisão	**Tempo(s)**	**Precisão**	**Tempo(s)**
SimpleDic	94,72	3,65	63,23	12,79
CALIG	92,13	12,78	92,42	44,79
CALIM	98,93	2,87	97,08	10,06
LangDetect	99,48	3,25	99,16	11,38

De acordo com o experimento acima, o LangDetect continua melhor com 0,55 e 2,08 pontos percentuais acima do CALIM para 6 e 21 idiomas respectivamente e o CALIM continua mais rápido que os demais em no mínimo 0,38 e 1,32 segundos para os mesmos grupos de idiomas. Com o experimento utilizando o corpus na versão *Full* ter-se-á a impressão do funcionamento dos algoritmos em documentos que assemelham a realidade da web em termos de formato (XML) e tamanho (maior quantidade de documentos, em quantidade e tamanho das sentenças).

Tabela 7. Os resultados médios com 6 e 21 idiomas (Europarl *Full* Corpus)

Métodos	**6 idiomas**		**21 idiomas**	
	Precisão	**Tempo(s)**	**Precisão**	**Tempo(s)**
SimpleDic	100,00	871,76	80,123	3051,16
CALIG	99,97	676,97	99,942	2369,40
CALIM	100,00	793,37	99,992	2776,79
LangDetect	100,00	910,32	99,993	3182,12

[13] Alemão, inglês, espanhol, francês, italiano e português.
[14] Todos os idiomas suportados pelo corpus Europarl já descritos anteriormente.

Segundo a tabela 7, que relata os resultados do experimento envolvendo o corpus na versão *Full*, que se assemelha com a realidade da web conforme já mencionado, o algoritmo CALIG obteve o melhor tempo, com tempo de processamento abaixo que os do CALIM e LangDetect em 14,67% e 25,63% respectivamente. Contudo, em termos de precisão para 6 e 21 idiomas, ficou abaixo em no mínimo 0,03 e 0,051 pontos percentuais em relação aos melhores (CALIM e LangDetect).

Em suma, os resultados sugerem que em termos de precisão e tempo de processamento, os algoritmos CALIM e CALIG obtiveram melhor desempenho sob os *corpora* na versão *Test* e *Full* do *Europarl*, com menores tempos de processamento e alta precisão, o que nos assegura a escolha do CALIM como método para classificação de idiomas oficial para nossa proposta.

2.6. Considerações Finais

Neste capítulo, foi apresentado em detalhes a classificação automática de idiomas em documentos textuais, envolvendo uma revisão da literatura dos trabalhos relacionados, uma proposta de um novo algoritmo híbrido de classificação. Além disso, foi apresentado o resultado de vários experimentos concernindo uma avaliação comparativa e homogênea do método de classificação proposto e outros algoritmos da literatura utilizando o mesmo ambiente (hardware, software e *corpus*). Concluiu-se que o algoritmo proposto revelou melhor desempenho combinando acurácia e tempo de processamento. A seguir, continuar-se-á a exposição neste mesmo modelo, focando-se no estudo da sumarização automática de textos.

3 Sumarização Automática de Textos

3

O erro acontece de vários modos, enquanto ser correto
é possível apenas de um modo.
Aristóteles

Tente de novo, fracasse novamente, porém, fracasse melhor.
Samuel Beckett

O rápido crescimento da Internet produziu uma enorme quantidade de informações disponíveis, especialmente no que se refere a documentos textuais (notícias, livros eletrônicos, artigos científicos, blogs, entre outros). Devido a esse grande volume de informações, tornou-se difícil uma mineração eficiente de informações úteis. Assim, faz-se necessário utilizar métodos automáticos para indexar, classificar, compreender e apresentar as informações de forma clara e concisa, permitindo que os usuários possam economizar tempo e recursos.

Uma solução é a utilização de técnicas de sumarização automática de textos, que é o processo de criação automática de uma versão compactada de um ou mais documentos (LINS, *et al.*, 2012), visando obter o "significado", ou melhor, a informação relevante contida dos documentos. Essencialmente, as *técnicas de sumarização* (TS) são classificadas como *extrativas* e *abstrativas* (LLORET & PALOMAR, 2012). TS extrativas produzem um subconjunto das sentenças mais importantes de um documento, exatamente como elas aparecem no documento original. Por outro lado, TS abstrativas produzem sumários visando auxiliar e melhorar a coerência entre as sentenças, eliminando redundâncias e deixando clara a relação entre as frases. Ele pode até produzir novas sentenças para o resumo. Neste ponto, um componente de geração automática de língua deve ser considerado (LINS, *et al.*, 2012).

A capacidade de sumarizar automaticamente um conteúdo é um trabalho complexo de mineração de texto. Spark-Jones (SPARCK-JONES, 1999) definiu sumarização como uma transformação redutora do texto de origem para um sumário ou resumo através da

compactação de conteúdo pela seleção e/ou generalização daquilo que é considerado importante no texto original. Pesquisas na área iniciaram-se em 1958 com Luhn (LUHN, 1958), que propôs analisar frequências e distribuições de palavras para calcular a importância das sentenças para criação de resumos. A necessidade cada vez maior de sumarização automática de documentos cativou mais e mais pesquisadores para a área (NENKOVA & MCKEOWN, 2012) (LLORET & PALOMAR, 2012) (CRUZ & URREA, 2005) (SPARCK-JONES, 1999).

Ao iniciar a pesquisa, encontra-se um número razoável de métodos propostos para selecionar as sentenças mais relevantes. Como em (TAKAMURA & OKUMURA, 2009) onde eles avaliam sentenças de acordo com modelos baseados em *cluster* ou grafos. A abordagem proposta por (WANG & LI, 2010) explora um algoritmo de agrupamento hierárquico incremental com o objetivo duplo de identificar grupos de sentenças que compartilham o mesmo conteúdo e atualizar os sumários ao longo do tempo. O LexRank (ERKAN & RADEV, 2004) propõe representar as correlações entre as sentenças por meio de um modelo de grafos. As frases mais relevantes são selecionadas de acordo com a centralidade do autovetor obtida através do bem conhecido algoritmo PageRank (BRIN & PAGE, 1998). Um esforço de investigação paralelo tem sido dedicado à formalização da tarefa de sumarização como um problema de máxima cobertura com grande quantidade de restrições baseadas na relevância da sentença dentro de cada documento, por exemplo, os trabalhos focados nas correlações entre palavras e sentenças são exemplos que a pesquisa está caminhando para este rumo, almejando chegar num patamar de obter não só uma medida de avaliação que informe o quão representativo é o sumário gerado, mas também se ele faz sentido como um todo, fator mais complexo e ainda em aberto.

Em virtude da quantidade e diferença entre os estudos realizados na área, sentiu-se a necessidade de avaliar sob a mesma perspectiva, antes de qualquer outra estratégia, ferramentas de sumarização disponibilizadas por pesquisadores na web, assim como outras de cunho comercial, e a partir daí, mensurar e criar algo que seja relevante. Assim, criou-se um módulo de extração de sumários das ferramentas de sumarização disponíveis via web, gerando assim um banco de sumários de diferentes ferramentas. Tais sumários foram gerados usando um corpus também criado neste trabalho, o qual visa preencher lacunas como: a falta de um corpus com qualidade textual, compreensível por qualquer área, que possua sumários *gold* (gerados por humanos), e relevante estatisticamente.

Diante disto, vislumbrou-se preencher essas lacunas com um corpus extraído a partir das notícias contidas no site CNN (www.cnn.com). A vantagem do uso deste novo corpus repousa sobre a qualidade textual e em torno dos destaques (*highlights*) oferecidos para cada texto, trata-se de um sumário relevante, contendo 3 ou 4 sentenças, fornecido pelo próprio editor. O corpus CNN na versão 1 abrangia 400 textos e era, possivelmente, um dos maiores *corpora* de testes existentes para área de sumarização na época.

Deste modo, devido à multiplicidade de ferramentas de sumarização existentes e à necessidade de uma avaliação correta das ferramentas sob o mesmo corpus e hardware, foi proposta uma abordagem para melhorar a compactação global, a de encontrar um meio para combinar de forma eficaz o resultado de várias ferramentas de sumarização. Este experimento revela um novo enfoque para a sumarização de texto que utiliza os resultados de diferentes ferramentas para gerar um resumo híbrido, que se pretende captar o melhor de cada uma das ferramentas através de um método adaptado do algoritmo *K-means*, melhor detalhado na seção 3.4.

3.1 Ferramentas de Sumarização Automática

Seis ferramentas de sumarização foram escolhidas para fornecer entrada para compor o sumário "híbrido" gerado no experimento aqui descrito. São elas: TextCompactor (Knowledge by Design, Inc., 2014), FreeSummarizer (Free Summarizer, 2014), Smmry (ELMAANI, 2009), WebSummaryzer (Context Discovery Inc., 2012), Interllexer (Intellexer Inc., 2014), e o Compendium (LLORET & PALOMAR, 2013). Todas apresentam características comuns: *blackbox* (código privado), pré-processamento usando separação individual de palavras (*tokenizer*) e sentenças (*splitter*).

3.1.1 TextCompactor

TextCompactor é uma ferramenta de sumarização disponível na web, criada por Keith Edyburn para a empresa Knowledge by Design, Inc. Ela é usada para ajudar os leitores na dificuldade de processar uma grande quantidade de informação. Para resumir o texto, ela calcula a frequência de cada palavra nas sentenças. Em seguida, a pontuação é calculada para cada frase com base na contagem da frequência obtida das palavras. As sentenças com maiores pontuações são consideradas as mais importantes. A ferramenta em questão funciona melhor com textos expositivos, como artigos, notícias, entre outros, não sendo recomendado seu uso com textos de ficção, i.e., peças teatrais, novelas, entre outros.

Seu funcionamento é simples necessita que o usuário envie um arquivo de texto, e o percentual de redução. As frases escolhidas não diferem do arquivo original. A limitação é não lidar com arquivos longos (superiores a 15 mil caracteres).

3.1.2 FreeSumarizer

Esta ferramenta cria um sumário extrativo baseado nas frequências das palavras. O serviço é gratuito. Ele permite que o usuário selecione a quantidade desejada de sentenças do resumo. Como o TextCompactor, frases escolhidas não são alteradas em relação ao texto original, assim como há limite de processamento para arquivos com 15 mil ou mais caracteres, também não é levado em consideração a estrutura do documento.

3.1.3 Smmry

Criada em 2009 por Amir Elmaani, tal ferramenta cria um resumo seguindo cinco etapas: 1) Seu algoritmo principal ordena as sentenças por importância; 2) Reorganiza o sumário para focar em um tópico; usando seleção de palavras-chaves; 3) Retira sentenças de transição; 4) Remove cláusulas desnecessárias; 5) Retira exemplos em excesso.

O algoritmo principal calcula a ocorrência de cada palavra no texto, depois associa palavras com os seus homólogos gramaticais. Em seguida, ordenam-se as sentenças através da soma de pontos das palavras contidas nelas. A ferramenta foi desenvolvida em PHP, funciona online e como uma API tendo como entrada arquivos de texto (txt) ou hipertexto (HTML), produzindo uma saída do mesmo tipo de arquivo. As sentenças de saída podem ser ligeiramente modificadas, como exemplo: frases de transição, apostos desnecessários, exemplos excessivos são removidos.

3.1.4 WebSummaryzer

WebSummarizer é uma aplicação desenvolvida pela empresa Context Discovery Inc. Ele suporta sumarização de conteúdo em inglês, francês, alemão e espanhol. O resumo é criado usando classificação de sentenças, e ele é apresentado no formato de esquema estruturado textual e no formato visual. O resumo nesses formatos são mapas interativos de conteúdo que os usuários podem navegar entre as palavras-chaves para ver instantaneamente os sumários importantes no contexto selecionado. É importante observar que nenhum dos resumos é criado ou revisado por humanos; todo o processo é automático. Há uma versão *online* de testes e uma API para o licenciamento. Esta ferramenta trabalha com entradas de vários tipos: texto simples, e-mails, doc, pdf, etc. A saída pode ser em formato textual. As

sentenças escolhidas são mantidas inalteradas. A ferramenta é capaz de processar arquivos grandes (de dimensão superior a 15 mil caracteres). A informação de tipo de documento não é utilizada no processamento.

3.1.5 Intellexer

A ferramenta comercial chamada Intellexer Document Sumarizer (Intellexer Inc., 2014) é uma aplicação *desktop* em duas versões diferentes: um de uso geral e outra profissional (Pro). Embora ambas as versões tenham a mesma interface de usuário e funcionalidade idênticas, a diferença entre elas reside nos algoritmos internos de funcionamento e pacotes de vocabulário incluídos.

A versão Professional possui as seguintes características:

- Afirma oferecer qualidade profissional de sumarização mesmo para textos complexos, como documentos para juristas, pesquisadores ou analistas de notícias.
- Adequado para documentos de propósitos específicos: tais como patentes, artigos científicos, análises econômicas, dentre outros.
- Ajustado para assuntos como: Geral, Patente, Ciência, Economia, Política, Direito, Saúde, Tecnologia, Desastres, Ecologia, Esportes, Inovação.
- Compatível com arquivos PDF, TXT, HTML/HTM, O DOC, PPT RTF, HTML, CHM, URL, DOCX, MHTML/MHT.
- Não possui limites para entrada ou saída.

Uma versão de demonstração válida por 30 dias está disponível no site do produto: http://summarizer.intellexer.com/summ_demo_v2.php.

3.1.6 Compendium

Esta aplicação (LLORET & PALOMAR, 2013) é uma ferramenta de sumarização capaz de gerar os tipos mais comuns de sumários. Com esta ferramenta o usuário pode gerar sumários extrativos e abstrativos a partir de um ou vários documentos, tanto focado em consultas como baseado em sentimento. As principais contribuições dela são:

1) Uso de vinculação textual para evitar informações redundantes nos resumos;
2) Combinação de estatísticas e técnicas baseadas em cognição para detectar informações relevantes; e
3) Geração de sumários orientados do modo abstrativo.

A aplicação foi desenvolvida por (LLORET & PALOMAR, 2013) e executa:

1) Uma análise de superfície linguística (*Tokenizer*, *Sentence Splitter*, *POS-tagging*, *stemming*, identificação de *stopwords*);
2) Detecção de redundância (usando para isso a técnica de vinculação textual);
3) Identificação de tópicos (identificar os principais tópicos de um documento);
4) Detecção de relevância (este estágio atribui um peso para cada frase, dependendo de como ela é relevante dentro do texto, usando O Princípio da Quantidade de Código);
5) Geração do sumário (as sentenças mais importantes, ou seja, a aquelas com notas mais altas, são selecionadas e extraídas).

A ferramenta funciona *online*, e tem um arquivo de texto como entrada e gera um arquivo similar na saída. As sentenças escolhidas não são modificadas. Apesar de não ser capaz de processar arquivos maiores que 15 mil caracteres, pode ainda receber mais de um arquivo como entrada para o resumo. A estrutura do documento não é considerada.

3.2 Exemplo de sumarização

A estratégia adotada para a geração de um novo sumário híbrido, tendo como entrada a saída das ferramentas descritas na última seção. A estratégia básica é comparar a frequência das sentenças na saída dos resumos e escolher as mais frequentes para comporem o resultado do esquema. Para exemplificar o resultado do esquema, apresenta-se um exemplo de uma notícia transcrita com as sentenças numeradas:

```
[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National Highway Traffic Safety Administration.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[5] That reduced clearance may mean the driver's foot could brush the side of the brake pedal when going from the accelerator to the brake, thus increasing stopping distances and the risk of a crash.
[6] Gas prices still slipping, survey says.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.
```

A seguir, segue um exemplo de sumário provido pelo Compendium:

```
[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs
because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[7] Gas prices still slipping, survey says Ford dealers will remove the carpet padding and
replace the left-side console trim panel free of charge, the NHTSA said.
```

Já o FreeSummarizer e Smmry produzem como resultado:

```
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel
free of charge, the NHTSA said.
```

O resultado do Interllexer é o sumário:

```
[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs
because of a potential problem affecting the brake pedal.
[5] The reduced clearance may mean the driver's foot could brush the side of the brake pedal
when going from the accelerator to the brake increasing stopping distances and the risk of a
crash.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel
free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact
the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.
```

O TextCompactor produziu o seguinte resumo:

```
[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs
because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[5] That reduced clearance may mean the driver's foot could brush the side of the brake pedal
when going from the accelerator to the brake, thus increasing stopping distances and the risk
of a crash.
[6] Gas prices still slipping, survey says.
```

E por fim, o WebSummarizer, proveu a seguinte saída:

```
[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National
Highway Traffic Safety Administration.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[7] Gas prices still slipping, survey says Ford dealers will remove the carpet padding and
replace the left-side console trim panel free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact
the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.
```

Pode-se observar que os sistemas escolhem algumas sentenças repetidas, mas também há divergências. O objetivo principal é manter as sentenças que são escolhidas pelos mais variados sistemas, assim como incluir as sentenças que estão aparecendo de forma esparsa entre os algoritmos analisados, utilizando uma variante do algoritmo *K-means*, descrito e exemplificado na seção 3.4.

3.3 Resultados Preliminares

O resumo de 4 sentenças obtido através do esquema de votação proposto foi este:

```
[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs
because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel
free of charge, the NHTSA said.
```

O resumo acima parece ser uma porção representativa do texto original. O artigo CNN analisado possui uma série de virtudes: é extremamente conciso, claro e objetivo. Além disso, ele fornece alguns destaques textuais (*highlights*), os quais se tratam de pontos importantes contidos no artigo em questão, em outras palavras, é um sumário disponibilizado pelo autor do texto. Abaixo, têm-se os destaques do artigo exposto acima:

```
The recall affects 8,266 Escape SUVs.
Mispositioned carpet can reduce clearance around the brake pedal.
Dealers will correct the problem free of charge.
```

Nota-se que cada uma das sentenças pode ser facilmente correlacionada para uma ou mais sentenças do artigo, abaixo, tem-se um exemplo de tal correlação:

```
[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National
Highway Traffic Safety Administration. There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed
outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel
free of charge, the NHTSA said.
```

Efetuar uma avaliação qualitativa em grandes conjuntos de dados é uma tarefa árdua; deste modo, procuraram-se métodos automáticos de avaliação de resumos. As pesquisas indicam o ROUGE (LIN, 2004) uma ferramenta que, dado um sumário a ser avaliado e um sumário *gold* (considerado ideal), retorna uma pontuação entre 0 e 1 para o sumário avaliado.

Em suma o ROUGE é um software de avaliação mais utilizado recentemente para a avaliação de sistemas de sumarização, com baixo esforço-administrativo (se comparado com uma avaliação manual) e consistência (evitando-se os erros humanos possíveis).

Baseado na medida BLEU, HAQUE, *et al.*, (2010), fortemente utilizada para a avaliação de sistemas de tradução automática, o ROUGE usa a abordagem de coocorrência de n-gramas, que consiste em verificar a média de quantas vezes cada conjunto de *n* palavras adjacentes se repetem em cada texto a ser avaliado, fornecendo como saída uma média de Cobertura, Precisão e F-measure (*Average_R, Average_P e Average_F*).

Usando os destaques textuais como sumários-modelo (chamados *gold-standard ou padrão-ouro*), fornecidos ao ROUGE para analisar os sumários obtidos pelas ferramentas pesquisadas e da proposta em questão com as seguintes configurações: 95% de intervalo de confiança, parâmetros de execução "-e .. /dados -c 95 -2 -1 -U -r 1.000 -n 4 -w 1.2 -a".

Tabela 8. Resultados do ROUGE para os resumos apresentados usando os destaques como *gold-standard.*

	Compend.	**FreeS**	**Interllexer**	**Summry**	**TextCom**	**WebSum**	**Proposed**
Average_R	0,54930	0,56338	0,53521	0,56338	0,47887	0,78873	0,74648
Average_P	0,30709	0,22857	0,24516	0,22857	0,28099	0,43750	0,38971
Average_F	0,39394	0,32520	0,33628	0,32520	0,35416	0,56281	0,51208

Usando as sentenças do artigo correlacionadas aos destaques textuais e fornecendo-as como sumário *gold-standard* ao ROUGE, seu cálculo produziu exatamente os mesmos resultados em relação à avaliação usando os destaques textuais para este exemplo. Este caso não é sempre verdadeiro, como pode ser demonstrado na seção a seguir. Uma explicação concisa para tal fato, é que o ROUGE usa medidas estatísticas baseadas em palavras, i.e. TF/IDF, Similaridade, etc., assim, mesmo possuindo duas sentenças diferentes, mas contendo palavras importantes em ambas, produzirão resultados semelhantes.

Entretanto, há a possibilidade da correlação entre o destaque e as sentenças do texto não terem exatamente um relacionamento biunívoco, pois pode haver um mapeamento sobrejetivo onde 1 sentença no destaque trate de assuntos que estão presentes em 2 ou mais sentenças do texto do artigo. O inverso também é verdadeiro.

3.4 Método Proposto

O método em si, consiste em selecionar sentenças para o sumário composto que foram escolhidas por, no mínimo, três leitores qualificados, seguindo uma média de 3,74 sentenças por sumário, sendo um mínimo de quatro e um máximo de seis sentenças à serem

selecionadas para o sumário híbrido. A estratégia adotada para descartar as sentenças excedentes ou fornecer o número mínimo foi a seguinte:

1. Calcular o score ROUGE de cada sumarizador usando o corpus CNN.
2. No melhor caso (há mais de seis sentenças no resumo híbrido), para cada sentença calcular seu peso como sendo a soma da pontuação dada pelo *Average_R* de cada um dos sumarizadores que escolheram essa frase. As sentenças com as menores pontuações serão descartadas.
3. No pior caso (não se obteve o mínimo de quatro sentenças no resumo composto), a estratégia adotada toma emprestado sentenças do sumarizador com a maior pontuação ROUGE, mas as sentenças escolhidas obedecem a uma distribuição espacial através dos índices das sentenças, para melhor representar todo o texto.

Tabela 9. Frequência das sentenças coincidentes escolhidas pelas ferramentas de sumarização.

Subject	**F=6**	**F=5**	**F=4**	**F=3**	**F=2**	**F=1**	**# Sentenças**	**# Textos**
Tecnologia	11	14	24	46	97	411	817	**25**
Turismo	5	11	21	29	45	126	1,625	**25**
Esportes	7	16	45	43	68	171	635	**25**
Negócios	7	16	32	47	62	136	457	**25**
Mundo	4	11	10	0	0	0	613	**25**
América latina	8	12	5	0	0	0	413	**25**
Europa	24	1	0	0	0	0	674	**25**
Oriente médio	35	25	15	0	0	0	1,700	**75**
Total	**101**	**106**	**152**	**165**	**272**	**844**	**6,934**	**250**

Como exemplo de como essas regras foram aplicadas, suponha-se que um resumo composto foi formado com duas sentenças: [17] (F=3) e [26] (F=2), i.e. a primeira foi escolhida por três diferentes ferramentas (F=3), já a segunda foi selecionada por duas (F=2). O sumarizador com a maior pontuação ROUGE selecionou cinco sentenças, a saber: [3] [7] [17] [25] [32]. O resumo composto deve ter um mínimo de quatro sentenças e duas ([17] e [26]) já foram escolhidas pela estratégia global, então o método precisa escolher duas sentenças dentre as divergentes do melhor resultado ROUGE: [3], [7], [25] e [32]. Tomando-se primeira sentença selecionada [17] como referência, as distâncias são: 14, 10, 8, 15. A partir da segunda frase selecionada [26], as distâncias são: 23, 19, 1, 6. A sentença [3] tem a maior distância do conjunto, assim é a candidata a ser incluída no resumo composto. Agora, as distâncias são recalculadas: a partir da sentença [3]: 4, 22, 29; a partir da [17]: 10, 8, 15; a partir de [26]: 19, 1, 6. A maior distância global é de sentença [32]. Assim, o sumário híbrido final teria as sentenças: [3], [17], [26], [32]. Tal estratégia de escolha de sentenças está relacionada com o método *K-means*.

3.5 Experimentos e Resultados

Visando uma melhor avaliação dos resultados obtidos a partir das ferramentas supracitadas e a estratégia proposta, seguiram-se algumas etapas em termos de desenvolvimento que merecem registro, diante das contribuições obtidas:

1. Um corpus usando notícias públicas do site CNN foi desenvolvido, na primeira versão, foram extraídos 250 textos distribuídos em categorias, tais como: tecnologia, viagens, esportes, negócios, notícias do mundo, da América Latina, Europa e Oriente Médio. Conforme abordado anteriormente, a vantagem intrínseca destes textos é sua qualidade de escrita, disponibilidade e a existência de um sumário provido pelo autor.
2. A quantificação, distribuição por classe, frequência de resultados coincidentes por classe podem ser vistos na Tabela 9.
3. Sumários obtidos de através de trabalhos relacionados, usando o corpus desenvolvido, para tanto, construiu-se um módulo extrator de sumários visando automatização da árdua tarefa. Neste módulo, ainda contém o método no qual viabiliza o sumário híbrido.

O resultado do cálculo ROUGE para as ferramentas de sumarização apresentadas e o sumarizador híbrido proposto, para os 250 textos do corpus CNN, usando os destaques dos artigos CNN como o ROUGE *gold-standard*, é mostrado na Tabela 10. Os resultados usando as sentenças dos artigos CNN correlacionadas com os destaques disponibilizados como ROUGE *gold-standard*, são mostrados na Tabela 11.

Tabela 10. Resultados do ROUGE tendo os destaques como padrão-ouro.

	Compend.	**FreeS**	**Interllexer**	**Proposed**	**Summry**	**TextCom**	**WebSum**
Average_R	0,51 ±0,16	0,51 ±0,13	0,52 ±0,16	0,56 ±0,15	0,55 ±0,15	0,58 ±0,17	0,52 ±0,14
Average_P	0,18 ±0,07	0,18 ±0,07	0,17 ±0,06	0,19 ±0,07	0,16 ±0,06	0,16 ±0,07	0,19 ±0,06
Average_F	0,26 ±0,09	0,25 ±0,07	0,27 ±0,09	0,25 ±0,08	0,24 ±0,08	0,26 ±0,08	0,27 ±0,08

Quando se compara as pontuações ROUGE, as ferramentas *Intellexer e WebSumarizer* fornecem os melhores resultados combinados (*F-measure* médio).

Tabela 11. Resultados do ROUGE tendo as sentenças correlacionadas aos destaques como padrão-ouro.

	Compend.	**FreeS**	**Interllexer**	**Proposed**	**Summry**	**TextCom**	**WebSum**
Average_R	0,55 ±0,20	0,56 ±0,18	0,58 ±0,21	0,65 ±0,21	0,62 ±0,20	0,65 ±0,23	0,57 ±0,20
Average_P	0,42 ±0,18	0,42 ±0,19	0,39 ±0,13	0,46 ±0,18	0,39 ±0,16	0,38 ±0,18	0,44 ±0,17
Average_F	0,45 ±0,15	0,46 ±0,15	0,45 ±0,13	0,52±0,16	0,46±0,16	0,45 ±0,16	0,49 ±0,17

Os resultados relativos ao ROUGE das sentenças que melhor correlacionam-se com os destaques, obteve o melhor resultado combinado (*Average_F*) o método proposto.

Deste modo, obteve-se um melhor entendimento do processo de sumarização em geral, a nova estratégia proposta para a sumarização de textos tomando várias ferramentas de sumarização como entrada, compondo os resultados utilizando método híbrido, envolvendo votação e uma variação do *k-means*, com objetivo de produzir sumários melhores, apresentou sucesso de acordo com o ROUGE. A estratégia proposta parece muito promissora em termos maximizar o nível de melhoria quantitativa (o ROUGE é estritamente quantitativo) da sumarização.

Um corpus de alta qualidade usando artigos de notícias da CNN foi desenvolvido para definir medidas justas de comparação. Ainda fornece-se uma avaliação sistemática de seis ferramentas de sumarização disponíveis.

3.6 Considerações Finais

Neste capítulo, apresentou-se a sumarização automática em documentos textuais em termos de uma revisão da literatura, isto é, uma análise sistemática de trabalhos relacionados. Além disso, foi introduzido um novo método de seleção de sentenças, o qual se mostrou competitivo nos experimentos e na avaliação homogênea realizada sob o mesmo corpus construído especificamente para este fim. A seguir, continuar-se-á a exposição neste mesmo modelo, focando um estudo mais aprofundado sobre sumarização, especificamente tratando de técnicas de sumarização extrativas.

4 Técnicas de Sumarização Extrativas

A imaginação é mais importante que o conhecimento.
Albert Einstein

As paixões ensinaram a razão aos homens.
William Shakespeare

Como já foi apresentado no capítulo anterior, a sumarização de textos é o processo de criação automática de um sumário a partir de um ou mais documentos, sendo dividida em dois tipos, extrativa e abstrativa. Atualmente, as técnicas de sumarização extrativas são mais utilizadas devido à facilidade de acesso, desenvolvimento e tempo de processamento. Diante do exposto, este capítulo concentra-se na descrição, implementação, experimentos e avaliação das técnicas de sumarização extrativas mais difundidas na literatura.

Para implementar métodos extrativos de sumarização, geralmente seguem-se três passos fundamentais (NENKOVA & MCKEOWN, 2012), enumerados abaixo:

1. Criação de uma representação intermediária do texto original;
2. Pontuação das sentenças;
3. Seleção das sentenças com maiores pontuações para o sumário.

No primeiro passo, cria-se uma representação do documento. Normalmente, ela divide o texto em parágrafos, sentenças, e *tokens*. Por vezes, pode-se utilizar algum componente de pré-processamento, tal como, e.g. remoção de *stopwords*. Na segunda etapa, tenta-se determinar quais as sentenças importantes para o documento, através de alguma medida quantitativa, atribuindo algum valor por sentença. Tal pontuação deve ser uma medida de quão significativa a sentença é para compreensão do texto como um todo. Na última etapa, ordenam-se as sentenças com base na pontuação obtida pela etapa anterior, define-se um limiar de corte e gera-se o sumário.

Este capítulo descreve 17 métodos de sumarização extrativa baseados em pontuação de sentenças, e algumas variações deles, amplamente utilizados e

referenciados na literatura, aplicados na sumarização de documentos únicos ou múltiplos nos últimos 10 anos. Os métodos supracitados são descritos e implementados.

Foram efetuadas avaliações quantitativa e qualitativa dos métodos utilizando mesmo ambiente (*hardware*, *software* e corpus). As medidas de precisão, cobertura e *f-measure* (BAEZA-YATES & RIBEIRO-NETO, 1999) fornecidas pelo ROUGE (LIN, 2004) foram utilizadas para realizar a avaliação quantitativa dos métodos estudados.

Já para avaliação "qualitativa", foram utilizados sumários definidos por humanos (quatro pesquisadores doutorandos). A partir dessas sentenças, contabilizou-se a quantidade de sentenças presentes em cada um dos métodos implementados, por definição, a avaliação não é apenas qualitativa, a melhor definição para ela é uma avaliação híbrida, pois possui aspectos qualitativos e quantitativos, evidenciando outra contribuição do trabalho.

Vale salientar que (LLORET & PALOMAR, 2012) e (NENKOVA & MCKEOWN, 2012) apresentam revisões atuais e abrangentes sobre a área sumarização de textos. Entretanto, eles não apresentam qualquer avaliação das técnicas que serão descritas aqui, fornecendo este capítulo mais uma contribuição visando o preenchimento desta importante lacuna de pesquisa. Além disso, algumas indicações sobre "Como podem ser melhorados os resultados pontuação de sentenças?" são apresentadas. A referência (FERREIRA, *et al.*, 2013) registram algumas das principais soluções para a questão em referência, tais como:

- Transformação morfológica do texto;
- Remoção de *stopwords*;
- Utilização de sinônimos;
- Resolução de correferências;
- Resolução de ambiguidade; e
- Redundância.

4.1 Revisitando os Métodos de Pontuação para Sumarização

A primeira referência para sumarização de textos usando pontuação de sentenças remonta a 1958 (LUHN, 1958) (LLORET & PALOMAR, 2009). Como já registrado outrora, o foco dessa área de pesquisa é guiado pela seguinte questão: como um sistema pode determinar quais sentenças são representativas do conteúdo de um texto específico? Em geral, três abordagens são utilizadas para pontuação: (i) palavras - atribuição de pontos para as palavras mais importantes; (ii) sentenças - verificando características de

sentenças tais como a sua posição no documento, semelhança com o título, entre outras; e (iii) grafos - analise da relação entre frases, designando pontuações. Apresentam-se a seguir detalhes sobre os principais métodos utilizados em cada uma das abordagens acima enumeradas.

4.1.1 Pontuação de Palavras

Os primeiros métodos em pontuação de sentenças foram baseados em palavras. Cada palavra recebe uma pontuação e o peso de cada frase é a soma de todas as pontuações das suas palavras constituintes. As abordagens contidas na literatura são descritas abaixo.

4.1.1.1 Frequência de Palavras

Como o nome sugere, as palavras mais frequentes no texto, recebem maiores pontuações (LUHN, 1958); (LLORET & PALOMAR, 2009); GUPTA, *et al.* (2011); (KULKARNI & PRASAD, 2010); ABUOBIEDA, *et al.* (2012). Em outras palavras, sentenças contendo as palavras mais frequentes em um documento terão uma chance maior de serem selecionadas para o sumário. A suposição é de que quanto maior a frequência de uma palavra no texto, mais provável é que ela indique o assunto principal do documento.

4.1.1.2 TF / IDF

A hipótese assumida por esta abordagem é que se existem "palavras mais específicas" em uma determinada sentença, então esta sentença é relativamente mais importante. As palavras em questão geralmente são substantivos, exceto para substantivos temporais ou adverbiais (MURDOCK, 2006; SATOSHI, *et al.*, 2001). Este algoritmo executa uma comparação entre a frequência termo (*tf*) num documento (neste caso, cada frase é tratada como um documento) e a frequência de documento (*df*), o que significa que o número de vezes que a palavra ocorre ao longo de todos os documentos. A pontuação TF / IDF é calculada da seguinte forma:

$$\frac{TF}{IDF(w)} = DN\left(\frac{\log(1+tf)}{\log(df)}\right) \tag{2}$$

onde DN é o número de documentos.

4.1.1.3 Maiúsculas

Este método atribui uma pontuação maior para palavras que contenham uma ou

mais letras maiúsculas, segundo PRASAD, *et al.* (2012). Pode ser um nome próprio, iniciais, palavras em destaque, entre outros. A pontuação é calculada da seguinte forma:

$$CPTW(j) = \frac{NCW(j)}{NTW(j)} \quad (3)$$

onde, *CPTW* = Razão do total de palavras com primeira letra maiúscula presentes na sentença com total de palavras presentes na sentença, *NCW* = Número de palavras com primeira letra maiúscula, e *NTW* = Número total de palavras presentes na sentença.

$$UCf = \frac{CPTW(j)}{Max\big(CPTW(j)\big)} \quad (4)$$

onde, *UCf* = é o coeficiente de pontuação utilizado neste método.

4.1.1.4 Nomes Próprios

Supõe-se que as sentenças que contêm um maior número de nomes próprios têm maior importância no texto; assim, elas são elegíveis a serem incluídas no sumário do documento (FATTAH & REN, 2009). Esta é uma especialização do método maiúsculas.

4.1.1.5 Coocorrência de Palavras

Neste caso, a coocorrência de palavras mede a chance de dois termos de um texto aparecer ao lado de outro em uma determinada ordem. Uma maneira de implementar esta medida é usando n-gramas (MARIÒO, *et al.*, 2006), que é uma sequência contígua de *n* itens de uma determinada sequência de texto. Em suma, ele dá maior pontuação para as sentenças que possuem mais frequentemente as coocorrências de palavras - LIU, *et al.* (2009); GUPTA, *et al.* (2011); (TONELLI & PIANTA, 2011).

4.1.1.6 Similaridade Léxica

Esta técnica baseia-se na suposição de que as sentenças importantes são identificadas pela ocorrência de palavras de mesmo significado ou outra relação semântica às outras, de importância reconhecida GUPTA, *et al.* (2011); (MURDOCK, 2006), e.g. Palavras mais frequentes ou nomes próprios.

4.1.2 Pontuação de Sentenças

Esta abordagem analisa as características da própria sentença e foi utilizada pela primeira vez em 1968 (EDMUNDSON, 1969) analisando a presença de palavras usadas como pontos de importância (*cue-phrases*) em sentenças. As principais técnicas que seguem esta linha estão descritas a seguir.

4.1.2.1 Ponto de Importância na Sentença (*Cue-phrases*)

Em geral, as sentenças que começam por "em suma", "conclui-se", "nossa pesquisa", "o documento descreve", além de frases fortes ou de efeito, e.g. "o melhor", "o mais importante", "de acordo com os estudos/resultados", "significativamente", "importante", "em particular", "dificilmente", "impossível", bem como termos de domínio específico em frases podem ser bons indicadores de quão significante é a sentença para um documento - GUPTA, *et al.* (2011); (KULKARNI & PRASAD, 2010); PRASAD, *et al.* (2012). A maior pontuação é atribuída para sentenças que contenham palavras ou frases com pontos de importância, utilizando a fórmula:

$$CP = \frac{CPS}{CPD} \tag{5}$$

onde, *CP* = é o coeficiente de pontos de importância (*cue-phrases*), *CPS* = são os pontos de importância na sentença, *CPD* = o total de pontos de importância no documento.

4.1.2.2 Dado Numérico na Sentença

Normalmente, a sentença que contém dados numéricos é importante e tem alta probabilidade de ser incluída no sumário do documento, segundo as referências (KULKARNI & PRASAD, 2010); (FATTAH & REN, 2009); ABUOBIEDA, *et al.*, (2012); PRASAD, *et al.* (2012). Esse tipo de frase normalmente se refere a algumas informações importantes, como data do evento, transação de dinheiro, porcentagem de ganho ou perda, entre outros.

4.1.2.3 Tamanho da Sentença

Este recurso é utilizado para penalizar sentenças muito curtas (FATTAH & REN, 2009) ou muito longas ABUOBIEDA, *et al.*, (2012), estas frases não são consideradas como uma seleção ideal para o sumário. Para calcular o tamanho da sentença o método usa o número de palavras contidas na frase. Além disso, (SATOSHI, *et al.*, 2001) penaliza sentenças que são mais curtas do que um determinado comprimento, i.e um limiar é definido. O primeiro caso pode ser calculado deste modo:

$$Pontuação(s) = Tamanho(Sentença) * TamanhoMédio(Sentenças) \tag{6}$$

Já a penalidade do segundo caso pode ser obtida usando a condição:

$$Penalidade(S_i) = \begin{cases} L_i & se\ (L_i > C) \\ L_i - C & senão \end{cases} \tag{7}$$

onde, L_i = tamanho da sentença *i* e *C* = tamanho definido como limiar pelo usuário.

4.1.2.4 Posição da Sentença

Existem muitas técnicas que usam a posição de sentença como um critério de pontuação (FATTAH & REN, 2009); (SATOSHI, *et al.*, 2001); (BARRERA & VERMA, 2012); ABUOBIEDA, *et al.*, (2012); GUPTA, *et al.* (2011). No trabalho ABUOBIEDA, *et al.*, (2012), a primeira sentença do parágrafo é considerada importante e uma forte candidata a ser incluída no resumo; GUPTA, *et al.* (2011) afirma que as primeiras frases dos parágrafos e palavras nos títulos e subtítulos são relevantes para a sumarização; O método proposto por (SATOSHI, *et al.*, 2001) atribui pontuação 1 (hum) para as primeiras *N* frases e 0 para as demais, onde *N* é um limiar sugerido para o número de sentenças.

Já a referência (FATTAH & REN, 2009) segue o mesmo princípio apresentado por SATOSHI e seus pares, além de assumir que as primeiras sentenças de um parágrafo são as mais importantes. As sentenças são ordenadas da seguinte maneira: a primeira frase no parágrafo recebe a pontuação 5/5, a segunda 4/5 e assim por diante, preterindo as sentenças que se localizam mais ao final do parágrafo. Por fim, (BARRERA & VERMA, 2012) explora o modelo de três posições. A primeira assume que as sentenças localizadas no início e no final do documento têm maior probabilidade de fornecer uma forte representatividade do conteúdo. A segunda prioriza apenas as partes que estão no início do texto. E a última usa sentenças que estejam tratando do mesmo tópico de chamada do documento (similar ao tópico contido no título ou subtítulo) para criação do sumário.

4.1.2.5 Centralidade da sentença

Trata-se do vocabulário que é tido na intersecção entre uma e as demais sentenças de um dado documento (FATTAH & REN, 2009); ABUOBIEDA, *et al.*, (2012); (KULKARNI & PRASAD, 2010). Esta técnica não utiliza tratamento semântico algum, limitando-se no nível léxico. Um caminho para obter tal medida é utilizar algoritmos de similaridade de sentenças, como por exemplo, o proposto (*Bleu*) por HAQUE e colegas (2010). A medida pode ser calculada da seguinte forma:

$$C(s) = \frac{K(s) \cap K(Os)}{K(s) \cup K(Os)} \tag{8}$$

onde, *K* é o conjunto com as palavras chave encontradas na sentença (*s*) e nas demais (*Os*).

4.1.2.6 Semelhança com o Título

Esta técnica trata-se do vocabulário coincidente entre uma sentença e o título do documento (SATOSHI, et al., 2001); (FATTAH & REN, 2009); (KULKARNI & PRASAD, 2010); ABUOBIEDA, *et al.*, (2012). Neste caso, sentenças similares ao título, devem conter palavras que são consideradas importantes. Uma forma simples de calcular essa pontuação é a seguinte:

$$Pontuação = \frac{Ntw}{T} \tag{9}$$

onde, *Ntw* é o número de palavras do título contidas na sentença e T é o número de palavras contidas no título.

4.1.3 Pontuação Baseada em Grafos

Nas abordagens baseadas em grafos, a pontuação é gerada pelo relacionamento entre as sentenças. Por exemplo, quando uma sentença se refere a outra, é gerado uma aresta com um peso associado entre as frases. Tais pesos são utilizados para gerar uma pontuação para cada sentença.

4.1.3.1 TextRank

Esta técnica trata-se de uma ordenação baseada no modelo de grafos para processamento textual (BARRERA & VERMA, 2012); (MIHALCEA & TARAU, 2004). Consiste na extração de palavras-chave importantes do texto e então determina um coeficiente de "importância" para estes itens. Sentenças e relacionamentos que contenham um maior número destas palavras-chave, obterão maiores pontuações.

4.1.3.2 *Bushy path*

A medida é definida simplesmente pelo número de arestas (links) conectando um nó (sentença) aos outros no grafo (FATTAH & REN, 2009).

4.1.3.3 Similaridade Agregada

Esta técnica gera uma medida da importância de uma sentença. Através da contagem do número de arestas conectando um nó do grafo (sentença) aos demais nós (*Bushy Path*), então efetua um somatório dos coeficientes (similaridades) presentes nas arestas (FATTAH & REN, 2009).

4.2 Experimentos: Materiais e Métodos Utilizados

Descreve-se nesta seção a metodologia para execução dos experimentos e avaliação dos algoritmos implementados.

4.2.1 Corpus

Conforme já apresentado na porção inicial do capítulo 3, desenvolveu-se neste trabalho um corpus de notícias gerais sobre diversos assuntos do mundo. Tais notícias foram extraídas de artigos públicos em páginas de agências de notícias, i.e. CNN. A versão estável do corpus contém 400 textos distribuídos em categorias, e.g. negócios, esportes, tecnologia, turismo, entre outros; e localidades como mundo, Estados Unidos, América Latina, Europa, entre outros.

A escolha da fonte da extração é a presença de textos de alta qualidade, concisos, de interesse geral, com temas atuais, claros e linguisticamente corretos. Mas a vantagem mais latente deste novo corpus é o sumário de boa qualidade contido em cada um dos textos extraídos, chamados de "destaques" (*highlights*). Tais componentes são providos pelos editores, contém em média entre três e quatro sentenças, aumentando sua importância, podendo-se utilizar este texto como sumário modelo (*gold-standard*) para fins de avaliação (LINS, *et al.*, 2012).

4.2.2 Especificação de Hardware e Software

Para executar estes experimentos, utilizou-se um computador Intel® Core i5-3320 vPro™ 2,60 GHz, 8 GB de memória com sistema operacional Microsoft® Windows™ 7 Professional 64 bits. Todos os algoritmos foram implementados na linguagem Java. Ferramentas de extração, geração de sumários de concorrentes e por humanos, usou-se C#.

4.2.3 Metodologia de Avaliação

Neste capítulo, têm-se duas metodologias de avaliação de resultados visando garantir uma avaliação de qualidade dos sumários obtidos.

4.2.3.1 Avaliação Quantitativa

Para esta avaliação continua-se utilizando o método descrito, apresentado e utilizado no capítulo 3, ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) (LIN, 2004) método amplamente utilizado para este propósito.

4.2.3.2 Avaliação Híbrida

Nesta avaliação, propõe-se uma hibridação de métodos, onde, aborda-se aspectos quantitativos e qualitativos. Quanto ao aspecto qualitativo, gerou-se uma ferramenta de anotação a qual, tem o objetivo de fornecer sumários gerados por humanos em corpora textuais para sumarização. Neste sentido, quatro pesquisadores (doutorandos) analisaram cada texto original do corpus de notícias CNN e selecionaram as sentenças nas quais, seriam indicadas para compor um sumário de qualidade. Já o aspecto quantitativo, simula a avaliação ROUGE, usando como *gold-stardard* os sumários gerados por humanos e quantifica o número de acertos das ferramentas com o sumário feito pelo homem. Tal quantificação é automatizada com módulo próprio para tal tarefa.

4.3 Avaliação do Desempenho da Sumarização

Descreve-se nesta seção (i) detalhes de implementação dos métodos e (ii) resultados da avaliação da performance de cada um.

4.3.1 Desenvolvimento

Todos os algoritmos detalhados na seção 4.1 foram implementados e neste momento, são apresentados os procedimentos gerais da atuação de cada um deles. Atentando ao fato de que há tarefas que são executadas por todos, e.g. a remoção das *stopwords (pré-execução),* e o somatório das pontuações quantificadas na execução por sentença, visando sua ordenação e seleção das melhores colocadas diante do limiar provido pelo usuário.

Frequência de palavra: conta todas as palavras do texto; Quantifica o número de aparições de cada palavra.

TF/IDF: calcula a formula apresentada na seção 4.1.1.2 para cada palavra do texto.

Maiúsculas: conta-se o número de palavras com letras maiúsculas no texto; calcula a formula apresentada na seção 4.1.1.3.

Nomes próprios: executa-se o *POS Tagging* usando Stanford CoreNLP visando selecionar os substantivos; contabiliza o número de substantivos que iniciam com letras maiúsculas.

Coocorrência de palavra: calcula a medida de n-gramas para um *n* variando entre dois, três e quatro.

Similaridade léxica: usa o WordNet para encontrar a similaridade entre palavras do texto e então aplica o algoritmo de frequência de palavra.

Pontos de importância (*Cue-phrases*): Carrega a lista de palavras que denotam um ponto de importância; quantifica o número total de itens da lista ocorrem no texto; calcula a fórmula apresentada na seção 4.1.2.1 para cada sentença.

Dados numéricos: utiliza expressões regulares para verificar se há dados numéricos presentes nas sentenças.

Tamanho da sentença: calcula o tamanho da sentença mais longa; penaliza sentenças maiores que 80% do tamanho da maior sentença; calcula o tamanho das demais sentenças.

Posição da sentença: Implementaram-se duas versões baseadas nas propostas de (FATTAH & REN, 2009) e (BARRERA & VERMA, 2012). Na primeira (1), as sentenças são ordenadas da seguinte maneira: a primeira sentença recebe o valor 5/5, a segunda 4/4 e assim sucessivamente. Na segunda (2) o procedimento ocorre com as últimas sentenças, a última recebendo 5/5, a penúltima 4/5 e assim por diante.

Centralidade da sentença: Foram implementadas duas versões, a primeira (1) baseada na medida Bleu (HAQUE e colegas (2010)) e em seguida verifica-se a similaridade entre sentenças. A segunda (2) é obtida com o resultado da formula apresentada na seção 4.1.2.5.

Semelhança com o título: obtém-se a pontuação através da implementação da formula apresentada na seção 4.1.2.5.

Similaridade agregada: criam-se arestas conectando uma sentença às demais; conta-se o número de arestas conectoras (*Bushy Path*); efetua-se um somatório das pontuações por sentença.

TextRank: utiliza o algoritmo provido no link https://github.com/turian/textrank.

Bush path: é calculado simplesmente usando o número de arestas conectando uma sentença as demais.

Visando facilitar a apresentação dos resultados, utilizar-se-ão algumas abreviações conforme apresentado nas tabelas a seguir.

Tabela 12. Descrição das abreviações dos resultados ROUGE.

Abreviação	**Descrição**
Average_R	Cobertura média
Average_P	Precisão média
Average_F	*F-measure* médio

Tabela 13. Abreviações dos algoritmos implementados.

Abreviação	Descrição
WordFreq	Frequência de palavras
TF/IDF	TF / IDF
UpperCase	Maiúsculas
ProperNoun	Nomes próprios
WordCo-oc	Coocorrência de palavras
LexicalSim	Similaridade léxica
Cue-phrases	Pontos de importância (*cue-phrases*)
NumericalData	Dados numéricos
SentenceLength	Tamanho da sentença
SentencePos_1	Posição da sentença (abordagem 1)
SentencePos_2	Posição da sentença (abordagem 2)
SentenceCent_1	Centralidade da sentença (abordagem 1)
SentenceCent_2	Centralidade da sentença (abordagem 2)
TitleResemb	Semelhança com o título
AggregateSim	Similaridade agregada
TextRank	TextRank
BushyPath	Bushy path

4.3.2 Resultados

Os resultados do cálculo ROUGE para cada um dos algoritmos, utilizando o corpus de notícias da CNN, usando como sumário padrão-ouro os sumários gerados por humanos, podem ser visualizados na tabela abaixo.

Tabela 14. Resultados do ROUGE para os algoritmos implementados.

Abreviação	Average_R	Average_P	Average_F
WordFreq	0,71(0,19)	0,35(0,13)	0,46(0,15)
TF/IDF	0,73(0,17)	0,35(0,12)	0,46(0,15)
UpperCase	0,64(0,19)	0,35(0,12)	0,44(0,12)
ProperNoun	0,64(0,20)	0,35(0,13)	0,45(0,15)
WordCo-oc	0,59(0,20)	0,33(0,13)	0,42(0,15)
LexicalSim	0,69(0,19)	0,35(0,13)	0,46(0,14)
Cue-phrases	0,50(0,22)	0,35(0,13)	0,40(0,14)
NumericalData	0,56(0,21)	0,36(0,13)	0,43(0,14)
SentenceLength	0,70(0,18)	0,33(0,12)	0,44(0,15)
SentencePos_1	0,61(0,22)	0,40(0,13)	0,47(0,15)
SentencePos_2	0,52(0,22)	0,36(0,13)	0,41(0,12)
SentenceCent_1	0,46(0,25)	0,37(0,16)	0,38(0,15)
SentenceCent_2	0,33(0,21)	0,31(0,13)	0,30(0,15)
TitleResemb	0,67(0,20)	0,36(0,12)	0,46(0,14)
AggregateSim	0,57(0,20)	0,34(0,12)	0,42(0,14)
TextRank	0,62(0,20)	0,34(0,12)	0,43(0,14)
BushyPath	0,56(0,20)	0,35(0,13)	0,42(0,14)

O resultado exibido é mostrado com o valor da medida associada à coluna (Cobertura, Precisão ou *F-measure*) e entre parênteses, encontra-se o desvio-padrão

registrado pelo ROUGE. Embora haja proximidade de valores nos resultados, alguns pontos merecem destaque.

- Os algoritmos *TF/IDF*, *WordFreq*, *SentenceLength* e *LexicalSim* alcançaram as melhores coberturas;
- Os algoritmos *SentencePos_1*, *SentenceCent_1*, *NumericalData*, *SentencePos_2* e *TitleResemb* atingiram os melhores valores de precisão;
- Os algoritmos *SentencePos_1*, *WordFreq*, *TF/IDF*, *LexicalSim* e *TitleResemb* registraram os melhores valores de *f-measure*;
- Os métodos de pontuação de palavra proporcionaram os melhores resultados dentre todos os algoritmos avaliados, ocupando três posições dentre os 5 melhores na avaliação realizada;
- O melhor algoritmo de pontuação palavra foi o *TF/IDF*;
- O melhor algoritmo de pontuação sentença foi o *SentencePos_1*;
- O melhor algoritmo de pontuação baseada em grafos foi o *TextRank*.

Como anteriormente mencionado, tem-se ainda uma avaliação híbrida descrita na seção 4.2.3.2, uma vez que o ROUGE é meramente quantitativo. Nela contabiliza-se o número de sentenças dos sumários gerados pelos algoritmos com o padrão-ouro gerado por humanos. A figura a seguir apresenta os resultados desta avaliação híbrida, denotando o número de sentenças corretas.

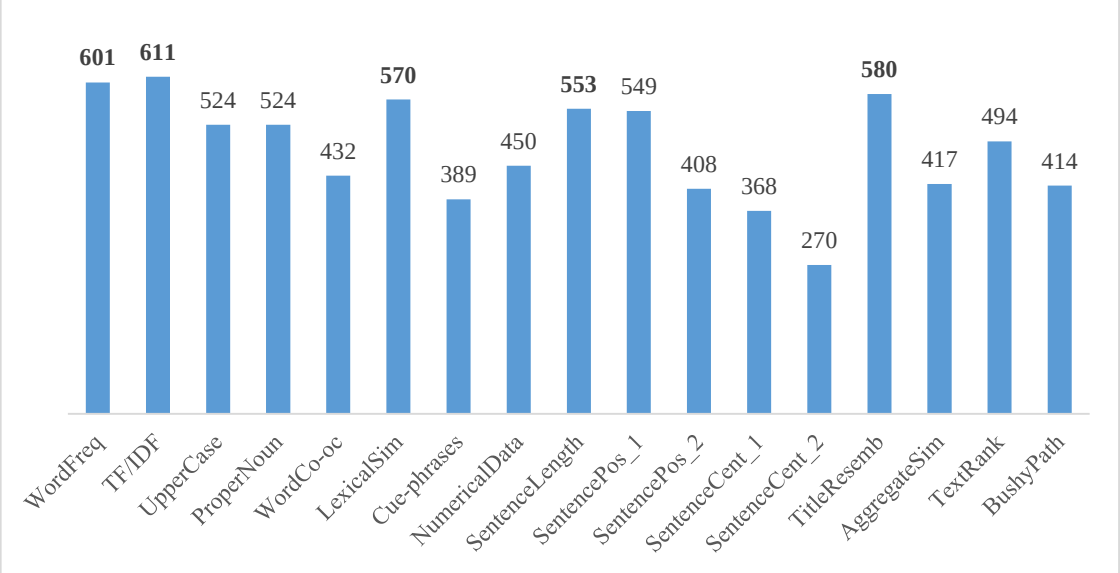

Figura 9. Número de sentenças corretas dos algoritmos implementados, segundo a avaliação híbrida.

As maiores pontuações foram obtidas pelos algoritmos *TF/IDF* (611), *WordFreq* (602), *TitleResemb* (580), *LexicalSim* (570) e *SentenceLength* (533), observa-se que os métodos baseados em pontuação de palavra continuam melhores (três dentre os cinco melhores) seguidos dos métodos baseados em pontuação de sentenças. Além disso, como importante meio de visualizar a efetividade dos algoritmos aqui estudados, registraram-se os tempos de processamento de cada um deles, apresentando-os na tabela a seguir.

Tabela 15. Tempo de execução utilizando 400 textos do corpus CNN.

Abreviação	**Tempo de execução *(s)***
WordFreq	13,986
TF/IDF	196,269
UpperCase	5,724
ProperNoun	25,723
WordCo-oc	20,490
LexicalSim	419,609
Cue-phrases	8,133
NumericalData	4,029
SentenceLength	4,820
SentencePos_1	4,122
SentencePos_2	4,292
SentenceCent_1	8,999
SentenceCent_2	47,267
TitleResemb	5,617
AggregateSim	7,708
TextRank	322,045
BushyPath	8,309

O resultado revela um importante fato, apesar dos bons resultados do algoritmo *TF/IDF*, existe uma maior necessidade de processamento em relação aos demais, denotando um alto tempo de processamento. Assim sendo tal algoritmo pode ser preterido na utilização de sumarização em tempo real para web, por exemplo, uma vez que seu tempo de execução da tarefa supera os algoritmos concorrentes *WordFreq* e *TitleResemb*, em exatamente 14,033 e 34,947 vezes, respectivamente. O algoritmo *NumericalData* foi o mais rápido, seguido do *SentencePos_1*, que por sua vez alcança bons resultados em tempo de processamento e nos experimentos quantitativos e qualitativos. O algoritmo *LexicalSim* apresenta relevantes resultados nos experimentos anteriores, entretanto neste, foi o mais lento entre todos os métodos.

4.3.3 Discussão

Considerando os resultados apresentados, pode-se afirmar que são relevantes diante dos documentos utilizados. Os artigos do corpus de notícias CNN são documentos muito bem estruturados, escritos, com corretude gramatical e vocabulário farto, o que denota os melhores resultados dos métodos de pontuação baseado em palavras (*WordFreq*, *TF/IDF* e *LexicalSim*).

Em geral, textos jornalísticos as primeiras e as últimas sentenças do documento apresentam impacto maior que as demais, as primeiras contendo o conteúdo da notícia sumarizado, e as últimas sentenças, emitem conclusões sobre o que foi tratado ao longo do artigo de forma clara e concisa. Tal fato explica os relevantes resultados dos algoritmos de pontuação baseado em sentenças (*SentencePos_1*, *SentencePos_2* e *SentenceLength*).

O algoritmo *TitleResemb* atingiu bons resultados devido aos jornalistas normalmente proverem títulos que possuem a principal informação do artigo em questão. No caso do algoritmo *SentenceCent_1*, registra-se boa precisão alcançada diante destes tipos de textos, que tendem a ser levemente redundantes, diante do reforço gradual no texto ao assunto foco da notícia. Já o algoritmo *LexicalSim* alcança bons resultados devido ao uso de sinônimos para escolha das sentenças, técnica conhecida não só da área jornalística, que visa evitar repetição de palavras, fazendo o autor demonstrar ao leitor um bom conhecimento de vocabulário e evitar a monotonia na leitura.

Por fim, em relação ao tempo de execução pode-se afirmar que o algoritmo *LexicalSim* é o mais lento devido ao tempo de processamento à consulta ao WordNet e ao cálculo de frequência efetuado nas palavras e em seus pares de semelhança, já os algoritmos *TF*/*IDF* e *SentenceCent*_2 tem um tempo de execução alto, devido as interações envolvendo equações necessárias para computação dos métodos.

Em geral, métodos de pontuação baseados em sentenças são mais rápidos. Isto é devido a eles utilizarem a estrutura da sentença, em detrimento aos métodos de pontuação baseado em palavras (que utilizam cálculos usando palavras) e em grafos (que criam um uma estrutura de grafo antes de executar o algoritmo), como exemplo tem-se o algoritmo *TextRank*, o qual não é rápido devido a necessidade de criação do grafo e execução da computação das palavras.

4.3.3.1 Como os resultados da pontuação de sentenças podem ser melhorados?

Os algoritmos de pontuação de sentenças estão chegando à maturidade, consequentemente, a comunidade científica atualmente está tentando melhorar seus

resultados assim como criando outros algoritmos. As seis estratégias mais utilizadas de acordo com a literatura são (NENKOVA & MCKEOWN, 2011; ORASAN, 2009): (i) transformação morfológica; (ii) *Stop words*; (iii) Similaridade semântica; (iv) Correferência; (v) Ambiguidade, e (vi) Redundância. A seguir explica-se cada uma das estratégias listadas acima e apresentam-se possíveis soluções.

4.3.3.1.1 Transformação morfológica

O trabalho de (ORASAN, 2009) aponta para três transformações morfológicas que melhoram os métodos de pontuação baseada em palavra.

Trunking: Ele mantém apenas os seis primeiros caracteres das palavras que são mantidos em uma tentativa de identificar os *tokens* derivados da mesma raiz.

Stemming: É uma transformação que remonta às raízes das palavras, ou seja, retira o plural "s" de substantivos, o "*ing*" de verbos, ou outros afixos. A raiz é um grupo natural de palavras com significado igual (ou similar). Após o processo de transformação, cada palavra é representada por sua raiz. Por exemplo, os verbos "*traveling*" e "*traveled*" ambos são transformados em "*travel*";

Lematization: Essa transformação identifica o lema de uma palavra. Por exemplo, ele mapeia os verbos no seu infinitivo e substantivos em sua forma singular. Assim, a forma da palavra precisa ser conhecida. Esta estratégia requer mais recursos do que os outros dois métodos. Ela pode lidar com palavras irregulares por meio de uma lista de exceções.

O trabalho citado (ORASAN, 2009) registra que as transformações listadas melhoram resultados de sumarização.

4.3.3.1.2 Stop words

O problema abordado aqui é a forma de lidar com as palavras com pouco significado para o texto, como artigos, conjunções e preposições. Além dessas, as palavras com altas e baixas frequências de ocorrência também são consideradas como *stop words*. É importante notar que algumas *stop words* podem ser significativas para sumarização textual, no entanto, e.g., algumas preposições podem se referir a temas importantes de texto (correferência).

Quase todos os sistemas de sumarização hoje efetuam tratamento de *stop words* de alguma forma (LLORET & PALOMAR, 2012; BARRERA & VERMA, 2012; WEI, 2012) e ABUOBIEDA, *et al.*, (2012).

4.3.3.1.3 Similaridade Semântica

Palavras de semântica semelhantes podem ser consideradas sinônimas. No entanto, as relações como hiperonímia e hiponímia também são importantes para melhorar o tratamento semântico. Relações de hiperonímia podem ocorrer quando as palavras estão relacionadas de algum nível de uma árvore semântica, e.g. "animal de estimação" e "cão", onde "cão" é um tipo de "animal de estimação", assim, eles estão relacionados.

No problema da pontuação de sentenças, palavras com semântica semelhantes poderiam ser consideradas como uma só, aumentando a importância relativa da palavra como conceito no texto.

Existem três abordagens em destaque para lidar com este problema. A primeira é usar relações do WordNet para verificar a similaridade entre duas palavras dadas. Neste, substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos cognitivos, cada um expressando um conceito distinto. Também possui hiperônimos e hipônimos. Uma visão geral de similaridade usando WordNet pode ser conferida no trabalho de Pedersen *et al.*, (2004). Alguns exemplos de sistemas de sumarização que usam WordNet são (LLORET & PALOMAR, 2013; BARRERA & VERMA, 2012) e outros como Zhang *et al.*, (2012) e Gupta *et al*., (2011).

A segunda abordagem que trata da semelhança semântica que é conhecida como cadeias léxicas (BARZILAY & ELHADAD, 1997). Esta abordagem explora intuitivamente que os temas são expressos usando não apenas uma única palavra, mas diferentes palavras relacionadas. Por exemplo, a ocorrência de palavras como "*car*", "*wheel*", "*seat*", "*passenger*" indica que o texto está relacionado com o tema automóvel, mesmo que cada uma das palavras não apareça com uma alta frequência no texto. Em outras palavras, esta estratégia agrupa palavras e os algoritmos de pontuação de sentenças, analisando temas ou conceitos, em vez de palavras isoladas. Tal abordagem é utilizada nos trabalhos de (WEI, 2012) e (GUPTA *et al*, 2011).

A última estratégia é a análise semântica latente DEERWESTER *et al.*, (1990). Esta é uma técnica não-supervisionada baseada na coocorrências de palavras para representar implicitamente a semântica textual, que tenta mapear quais palavras geralmente aparecem juntas HACHEY *et al.*, (2006).

4.3.3.1.4 Correferência

Correferência é o processo de combinar todas as referências à mesma entidade em um documento, independentemente da forma sintática da referência. Geralmente

corresponde a um substantivo, um sintagma nominal ou pronome. Alguns trabalhos têm demonstrado que a resolução de correferência pode ser usada para melhorar substancialmente os sistemas de sumarização que dependem de recursos de frequência de palavras (NENKOVA & MCKEOWN, 2011).

Um exemplo simples é a utilização de referência pronominal. Por exemplo, "*John will travel tomorrow. He bought the ticket yesterday.*" Neste caso, o pronome "*he*" se refere a "*John*". Assim, se as palavras são classificadas em conjunto, elas podem ser mais significativas. Este tipo de análise não é amplamente utilizada em sistemas de sumarização devido aos problemas de desempenho e de precisão.

4.3.3.1.5 Ambiguidade

Ambiguidade, também conhecida como polissemia, ocorre quando a mesma palavra pode ter significados diferentes em contextos diferentes. Por exemplo, "*apple*" pode significar uma fruta ou uma empresa de informática. Assim, os algoritmos de pontuação de sentenças podem atribuir valores mais altos para algumas palavras de forma inadequada. Cadeias léxicas podem resolver este tipo de problema. Duas questões fundamentais devem ser levadas em consideração no contexto de sumarização, normalmente em sumarização de documento único, palavras estão no mesmo contexto. Neste caso, a probabilidade da ocorrência de ambiguidade é baixa. Por outro lado, no contexto de sumarização de múltiplos documentos, tal problema pode ocorrer, mas resolver a ambiguidade pode aumentar problemas relacionados ao desempenho da aplicação.

4.3.3.1.6 Redundância

Ao contrário dos problemas apresentados anteriormente, a redundância está relacionada com frases e não apenas palavras. Redundância ocorre quando várias sentenças têm o mesmo conteúdo. Em geral, ela é percebida como imprópria, por causa da utilização de palavras duplicadas ou desnecessárias, principalmente em sumários.

As duas técnicas que são comumente utilizadas para tratar este problema são:

Fusão de sentenças: esta é a tarefa de retirar duas sentenças que contenham alguma informação sobreposta, mas que também contenham fragmentos que são diferentes, e produzindo uma frase que transmite as informações em comum entre as duas sentenças KRAHMER *et al.*, (2008).

Vinculação Textual: Trata-se de determinar se o significado de um trecho de texto (uma hipótese) pode ser inferido por um outro texto (GLICKMAN, 2009). A identificação

dessas relações de vínculo ajuda a um sistema de sumarização evitar redundância incorporada nos sumários gerados.

Estas técnicas são usadas, principalmente para sumarização abstrativa, mas elas podem ser perfeitamente adaptadas para a sumarização extrativa.

Por fim, as contribuições descritas aqui denotam relevantes caminhos para área de sumarização textual, além de implementar muitas estratégias da área encontradas na literatura nos últimos dez anos. Apresentam-se os cinco melhores resultados utilizando o avaliador quantitativo mais utilizado pela área (ROUGE) e ainda propõe-se um procedimento de avaliação híbrido, que une características qualitativas e quantitativas. Ambas as avaliações denotam coincidentemente como melhores algoritmos, estas quatro técnicas: frequência de palavra (*WordFreq*), *TF/IDF*, similaridade léxica (*LexicalSim*), e tamanho de sentenças (*SentenceLength*). A técnica *Text Rank* foi também destacada provendo bons resultados diante das técnicas baseadas em grafos. Os resultados providos pelo ROUGE para a avaliação quantitativa dos sumarizadores foi semelhante ao obtido pelo método híbrido. A computação do *TF/IDF* é mais intensa computacionalmente do que os demais métodos analisados. As técnicas de frequência de palavras e de tamanho de sentença provêm as melhores performances visando custo benefício, escolhendo melhores sentenças candidatas ao sumário num menor tempo de execução. Estratégias de composição dos resultados obtidos visando talvez, melhores sumários estão sendo investigadas e podem compor um bom artigo para publicação em um futuro próximo.

4.4 Considerações Finais

Neste capítulo, várias técnicas de sumarização extrativa foram apresentadas e discutidas. Tal análise sistemática de trabalhos relacionados promoveu a implementação de 17 diferentes algoritmos que foram categorizados em três grupos: algoritmos de pontuação baseada em palavras, baseado em sentenças e baseados em grafos.

Foram apresentados resultados experimentais, seguidos por discussões e uma análise mais detalhada da proposta de sumarização extrativa híbrida. Os resultados obtidos foram discutidos à luz dos aspectos quantitativos e qualitativos da solução proposta.

No próximo capítulo, apresenta-se a Sumarização Independente de Idiomas em detalhes, incluindo sua arquitetura geral, a integração dos módulos descritos nos capítulos anteriores, além de outros necessários, como o de tradução. Os experimentos e resultados para diferentes idiomas ajudam a dar a noção de independência defendida nesta tese.

5 Sumarização Independente de Idioma

5

Não sabendo que era impossível, foi lá e fez.
Jean Cocteau

Pagai o mal com o bem, porque o amor é vitorioso no ataque
e invulnerável na defesa.
Lao-Tsé

Chegando a este ponto, muito já se fez conforme pode ser visto nos capítulos anteriores, entretanto para o efetivo cumprimento da atividade fim da tese, é necessário o atendimento às demandas apresentadas no capítulo 1, que visam à utilização de métodos independentes de linguagem para entender, classificar e apresentar, de forma clara e concisa as informações existentes em diferentes idiomas, economizando recursos e tempo dos usuários. A sumarização independente de idioma aponta como uma solução viável para a necessidade supracitada, entre outras, como por exemplo, de uso seria na classificação de documentos, eliminando porções irrelevantes do texto, visando a criação de um sumário conciso para classificação de conteúdo em bibliotecas digitais. A maioria das técnicas de sumarização automática de documentos foi desenvolvida para o idioma inglês, deixando-se uma lacuna importante a ser preenchida por esta tese.

Neste sentido este capítulo apresenta uma proposta de plataforma para sumarização independente de idioma que combina técnicas para identificação de idiomas, tradução automática de conteúdo e sumarização. De forma sucinta, tal plataforma inicialmente efetua um pré-processamento do texto de entrada tornando-o compatível com os demais módulos. Logo após, identifica-se o idioma do documento, o resultado desta classificação é requisito para a condição de encaminhamento direto para o módulo de sumarização, caso idioma seja inglês, caso contrário para o módulo de tradução. Sendo o documento em inglês o sumário é gerado seguindo os trâmites apresentados no capítulo 4, entretanto caso seja de outra língua, o processamento é direcionado para o módulo de tradução, que traduz para o inglês e, em seguida, executa a sumarização. Os resultados são depois analisados para se obter o resumo final.

Outros trabalhos são encontrados na literatura visando realizar sumarização multilíngua. MEAD, por Radev e colegas (RADEV, *et al.*, 2004), faz uso de 8 algoritmos de sumarização multilíngua, sendo avaliado nos idiomas chinês e inglês. Evans e seus colaboradores (EVANS, MCKEOWN, & KLAVANS, 2005) usam semelhança e agrupamento de sentenças como estratégia de sumarização e seu trabalho é focado nos idiomas árabe e inglês. Roark e Fisher (ROARK & FISHER, 2005) utiliza aprendizagem de máquina para obter uma classificação de sentenças de forma supervisionada. Na referência (ROARK & FISHER, 2005) descreve-se um experimento com documentos traduzidos que tem algumas semelhanças com a estratégia proposta nesta tese. No entanto, o trabalho não menciona explicitamente o número de línguas suportadas, nem traz a ideia de ter mais de uma tradução do mesmo documento como uma forma de compensar o ruído adicionado pelo processo de tradução proposto aqui. Litvak, Last e Friedman (LITVAK, LAST, & FRIEDMAN, 2010), mais recentemente, usam algoritmos genéticos na tarefa de geração do resumo. Similarmente aos demais trabalhos aqui mencionados também suportam apenas duas línguas (inglês e hebraico). Gupta (GUPTA V. , 2013) utiliza um algoritmo híbrido para sumarização, com suporte a documentos dos idiomas / dialetos Hindi e Punjabi.

Infelizmente, nenhuma das referências listadas acima oferece elementos para testar seu desempenho em um conjunto de dados ou corpus comum dificultando uma análise ou avaliação de desempenho imparcial. Apesar disso, eles se concentram em um conjunto disjunto de idiomas (árabe, chinês, hebraico, hindi e punjabi) em relação aos seus continentes de origem, enquanto que aqui o foco são as línguas faladas na União Europeia. Como os trabalhos apenas tratam idiomas específicos: não há módulo de identificação de linguagem em suas soluções.

5.1 Descrição da arquitetura

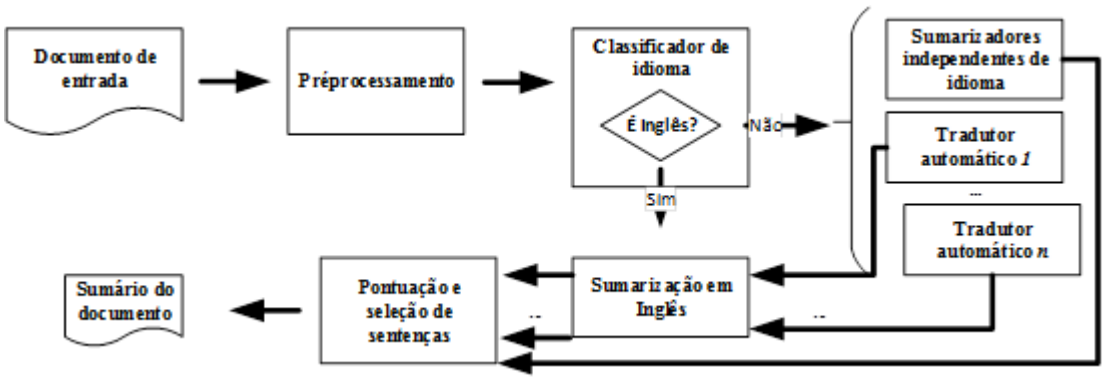

Figura 10. Arquitetura geral da plataforma.

Esta seção apresenta os detalhes dos principais módulos da Plataforma de sumarização independente de idioma, na Figura 10 contém um esboça a arquitetura geral. Na tarefa de pré-processamento remove-se as partes não-textuais do documento, efetua-se o tratamento textual (tokenização, separação de sentenças e indexação) fazendo com que casa frase tenha um índice.

Na identificação é reconhecido o idioma do documento. Se o idioma for inglês, submete-se o texto ao módulo de sumarização extrativa, que seleciona as frases mais significativas a partir do original utilizando os métodos pontuação apresentados no capítulo 4; caso seja outro idioma, submete-se o documento ao módulo composto por algoritmos de sumarização independentes de idioma e por variados tradutores automáticos que podem traduzir o texto original para inglês mantendo sua indexação.

Como o processo de tradução automática pode adicionar ruído ao texto traduzido, a utilização de mais de uma ferramenta de tradução pode compensar tais ruídos. As versões do texto traduzido são submetidas ao módulo de sumarização extrativa produzindo para cada entrada um subconjunto de sentenças traduzidas eleitas para o sumário, mantendo a compatibilidade dos índices entre o documento original e os traduzidos.

Os resumos são analisados pelo módulo de pontuação e seleção de sentenças, que irá produzir um conjunto de sentenças que correspondem ao resumo, buscando as frases no texto original pelo casamento dos índices. Gerando-se na saída um resumo com o idioma original do documento. A seguir os módulos são detalhados.

5.1.1 Pré-processamento

Atividade tecnicamente simples, onde efetuam-se a retirada de itens não textuais, como figuras e vídeos de notícias, assim como a separação das sentenças via expressão regular. Um exemplo pode ser visto abaixo.

```
Entrada = "Olá mundo! Como está você? Eu estou bem. Esta é uma sentença difícil porque
estou usando I.D.
Novas linhas também precisam ser aceitas. Números não poderiam causar uma quebra de
sentença, como 9.50.";

Regex rx = new Regex(@"(\S.+?[.!?])(?=\s+|$)");
//… código para separação de sentenças …

Resultado:
Olá mundo!
Como está você?
Eu estou bem.
Esta é uma sentença difícil porque estou usando I.D.
Novas linhas também precisam ser aceitas.
Números não poderiam causar uma quebra de sentença, como 9.50.
```

5.1.2 Identificação de Idiomas

O método escolhido para ser utilizado neste módulo da plataforma é o CALIM descrito em detalhes no capítulo 2, tendo como maior justificativa a escolha, seu custo-benefício envolvendo alta precisão e baixo tempo de processamento.

Experimentos envolvendo o *Europarl v7 "full" corpus* (KOEHN, 2005) com cerca de 60,000 documentos distribuídos aleatoriamente entre os 21 idiomas da comunidade Europeia que foram utilizados nos testes alcançando a taxa de 99,992% de precisão. Melhorias foram implementadas incluindo suporte outros idiomas (árabe, hebreu, hindu e koreano) usando identificação baseada em alfabeto alcançando 100% de precisão no reconhecimento.

5.1.3 Tradução Automática Intermediária

Este módulo executa um processo de tradução automática intermediária usando mais de uma ferramenta de tradução. Ele requer como entrada o texto a ser traduzido, além do idioma de origem e destino, aumentando assim a importância do módulo de identificação de idiomas. Na tarefa são utilizadas APIs disponíveis para executar as traduções, tais como o Microsoft Translator (MICROSOFT, 2014) e Google Translate (GOOGLE, 2012) APIs.

Devido à possibilidade de adição de ruídos, utilizam-se as duas ferramentas de tradução, afim de verificar quão consistente é o resultado de uma ou outra, visando o objetivo maior que é a sumarização de qualidade. Algumas diferenças de utilização das ferramentas proporcionaram problemas ao longo das pesquisas, no caso da ferramenta da Microsoft, que apesar de conter restrições, continua disponível sem cobranças para um número limitado de caracteres por requisição (cerca de 10 mil) e por mês (cerca de 2 milhões). Já no caso da ferramenta do Google, a tradução é cobrada, 20USD a cada 1 milhão de caracteres traduzidos ou de forma proporcional, fato que diminui as chances de utilização da ferramenta em um futuro produto.

Tal função é uma contribuição fundamental, que agrega à plataforma a condição de utilizar os algoritmos de sumarização, antes, dependentes do idioma inglês (por exemplo, pontos de importância (*cue-phrases*) e nomes próprios), agora, podendo suportar outros idiomas. O fato da independência de idiomas encontra-se ainda nos algoritmos de sumarização estatísticos, nos quais não requerem idioma predefinido para sua execução.

Inicialmente foi necessário verificar se ocorriam mudanças nas sentenças após o processo de tradução, por isso, a partir do corpus espanhol CNN, foi realizada a tradução por diferentes APIs, e a verificação foi realizada, retornando que as sentenças são alteradas, mas, o seu índice é mantido, como pode ser visto no exemplo abaixo.

(a) Conteúdo original de uma notícia em espanhol.

```
[1] (CNNMéxico) — El jamaicano Usain Bolt, que consiguió este domingo su tercera medalla
de oro en Moscú, su octava medalla de oro en campeonatos del mundo y la décima en total,
se dijo orgulloso de sí mismo y anunció que seguirá trabajando "para dominar tanto tiempo
como sea posible".
[2] "Da gusto vencer", dijo, luego de ganar con el equipo jamaicano el primer lugar en la
carrera de relevos 4x100, según EFE.
[3] "Para eso he estado entrenando.
[4] He trabajado mucho y muy duro, superando todos los obstáculos que he ido encontrando
en mi camino.
[5] Estoy orgulloso de mí mismo y voy a seguir trabajando para dominar tanto tiempo como
sea posible".
[6] Bolt no tiene claros sus planes inmediatos para cerrar la temporada.
[7] "No estoy en la forma en que me gustaría estar, así que vamos a ver qué pasa con la
final de la Diamond League", dijo, según EFE.
[8] Cuestionado sobre si estaba pensando en tomarse un descanso la próxima temporada,
dijo: "No lo sé todavía, mi entrenador (Glen Mills) decidirá".
[9] El seis veces campeón olímpico enmarcó sus triunfos dentro del desarrollo del atletismo
en su país.
[10] "Los talentos de la velocidad pueden surgir en cualquier sitio, no solo en Jamaica,
pero el atletismo se está haciendo cada vez más grande en Jamaica y creo que pronto veremos
muchos otros talentos", dijo.
[11] Con la presea dorada de este domingo, Bolt se convirtió en el atleta más premiado de
la historia, superando a la leyenda estadounidense de las pistas Carl Lewis, el Hijo del
Viento.
[12] Bolt y Lewis acumulan las mismas medallas en mundiales de atletismo, 10.
[13] Sin embargo, el jamaiquino suma 8 de oro y 2 de plata, con un valor superior a las
de Lewis que acumuló 8 de oro, 1 de plata y 1 de bronce.
[14] En días anteriores, durante las actividades del Mundial de Atletismo en Moscú, Rusia,
Bolt recuperó su corona en los 100 metros y también se posicionó como el mejor en los 200.
[15] La última victoria de Bolt, al igual que la de sus compatriotas, llega como aire
fresco para el deporte en Jamaica, sacudido en los últimos meses por escándalos de dopaje
como el del velocista Asafa Powell.
```

(b) Tradução efetuada pela API Microsoft Tradutor.

```
[1] (CNNMéxico) - the Jamaican Usain Bolt, who won on Sunday their third gold medal in
Moscow, its eighth gold medal in the World Championships and tenth overall, said proud of
itself and announced that it will continue to work "to dominate as long as possible".
[2] "taste defeat," said, after winning first place with the Jamaican team in the 4 x 100,
according to EFE relay race.
[3] "for that I've been training.
[4] I have worked very hard and very hard, overcoming all the obstacles that I have been
finding my way."
[5] I am proud of myself and I'm going to continue to dominate as long as possible".
[6] Bolt has no clear their immediate plans to close the season.
[7] "I am not in the form in which I'd like to be, so let's see what happens with the end
of the Diamond League', said, according to EFE.
[8] Cuestionado whether was thinking about take a break next season, said: "I don't know
yet, my coach (Glen Mills) will decide".
[9] six-time Olympic champion framed their triumphs in the development of Athletics in
the country.
[10] "the talents of speed can arise anywhere, not only in Jamaica, but athletics is
becoming increasingly large in Jamaica and I believe that we will soon see many other
talents," said.
[11] with the gold medal on Sunday, Bolt became the most awarded athlete of history,
beating American tracks legend Carl Lewis, the son of the wind.
[12] Bolt and Lewis accumulate the same medals in World Athletics, 10.
[13] However, the Jamaican added 8 gold and 2 silver, with a value greater than the Lewis
who amassed 8 gold, 1 silver and 1 bronze.
[14] in previous days, during the activities of the Athletics World Cup in Moscow, Russia,
Bolt regained his Crown in the 100 meters and also ranked as the best in the 200.
[15] the last victory of Bolt, as well as of their compatriots, arrives as fresh air for
the sport in Jamaica, shaken in recent months by scandals of doping as the sprinter Asafa
Powell.
```

(c) Tradução efetuada pela API Google Translate.

```
[1] (CNNMéxico) - Jamaican Usain Bolt, who won Sunday its third gold medal in Moscow, his
eighth gold medal in the World Championships and the tenth in total, said proud of himself
and announced that it will continue to work "to master as long as possible."
[2] "It's nice to win," he said after winning the Jamaican team to first place in the
4x100 relay race, according to EFE.
[3] "That's what I've been training.
[4] I have worked long and hard, overcoming all obstacles that I found my way.
[5] I am proud of myself and I will continue working to master as long as possible. "
[6] Bolt has no clear immediate plans to close the season.
[7] "I'm not in the way that I would be, so we'll see what happens with the final of the
Diamond League," he said, according to EFE.
[8] Asked if he was thinking of taking a break next season, said: "I do not know yet, my
coach (Glen Mills) decide."
[9] The six-time Olympic champion framed his triumphs in the development of athletics in
the country.
[10] "The talents of speed can arise anywhere, not only in Jamaica, but the track is
getting bigger in Jamaica once and I think we will soon see many other talents," he said.
[11] With the gold medal on Sunday, Bolt became the most decorated athlete in history,
surpassing the US track legend Carl Lewis, the Son of Wind.
[12] Lewis Bolt and accumulate the same medals at world championships 10.
[13] However, the Jamaican sum 8 gold and 2 silver, with an excess of Lewis accumulated
value 8 gold, 1 silver and 1 bronze.
[14] In earlier days, during the Global Athletics in Moscow, Russia, Bolt regained his
crown in the 100 meters and also ranked as the best in the 200.
[15] Bolt's last victory, like that of his countrymen, comes as fresh air for the sport
in Jamaica, rocked in recent months by doping scandals like the sprinter Asafa Powell.
```

Como pode ser visto no exemplo anterior, realmente há diferenças de tradução, muito embora não se encontre problemas entre sentenças, há ruídos de tradução nos quais não obedecem a gramática formal inglesa, como na utilização de preposições ou adjetivos (por exemplo `The last victory of Bolt` invés de `The last Bolt victory`) além de um tradutor manter a estrutura original da sentença (Microsoft) em relação ao outro (Google), um exemplo disto é como o Google usa de características da gramática inglesa (por exemplo `Bolt's`), o que faz da sentença se distanciar da estrutura da sentença original, como sumarização trabalha com palavras, o modelo do tradutor Microsoft é valorizado por manter tal estrutura.

Observa-se que apesar destes ruídos, não houve em todas as observações efetuadas, movimentação de um trecho texto de uma sentença para outra. Assim a indicação do uso de índices nas frases é interessante e deve ser mantida, o que ajuda na estratégia proposta, que diz respeito ao traduzir o conteúdo original para inglês, para executar o processo de sumarização e, finalmente, com os resultados, realizar um mapeamento bijetivo entre as sentenças do resumo obtido e o documento original, mostrando no final, um resumo em língua original do documento, sem qualquer ruído.

Neste sentido, para fins de avaliação do processo de tradução, realizou-se um experimento com o objetivo de medir a sensibilidade da sumarização após o processo de tradução, as fases do experimento e suas descrições podem ser vistos descrito abaixo:

Fase 1: Selecionar um grupo de notícias em espanhol (400 documentos);

Fase 2: Gerar sumários de 4 sentenças deste corpus para cada um dos 17 métodos de sumarização e para cada um dos tradutores implementados;

Fase 3: Verificar a correlação dos índices das sentenças dos sumários gerados pelos métodos de sumarização por tradutor com os sumários padrão-ouro, gerados por humanos, a fim de verificar a precisão, ou, coeficiente de sensibilidade, que diz o quão sensível é o processo de sumarização diante da tradução intermediária. Quanto mais alto for o valor, menores foram as divergências de sentenças entre os sumários gerados e padrão-ouro, na contramão, quanto menor o valor, maiores foram as diferenças.

Ao final dos experimentos obtiveram-se os resultados mostrados na Tabela 16. Para facilitar o entendimento, segue-se o mesmo padrão de nomenclatura utilizado nos experimentos do Capítulo 4.

Tabela 16. Coeficiente de confiança dos tradutores.

Abreviação	Método de sumarização	Google	Microsoft
WordFreq	Frequência de palavras	50,5834	**55,2499**
TF/IDF	TF / IDF	55,7913	**61,7913**
UpperCase	Maiúsculas	57,2916	**61,7497**
ProperNoun	Nomes próprios	54,1666	**58,3747**
WordCo-oc	Coocorrência de palavras	45,3751	**47,2918**
LexicalSim	Similaridade léxica	47,8750	**52,5834**
Cue-phrases	Pontos de importância (*cue-phrases*)	**33,9588**	33,0003
NumericalData	Dados numéricos	50,1667	**54,2916**
SentenceLength	Tamanho da sentença	56,8747	**64,2914**
SentencePos_1	Posição da sentença (abordagem 1)	79,1244	**81,2911**
SentencePos_2	Posição da sentença (abordagem 2)	34,5420	**41,1668**
SentenceCent_1	Centralidade da sentença (abordagem 1)	53,9166	**62,7495**
SentenceCent_2	Centralidade da sentença (abordagem 2)	44,0836	**50,2915**
TitleResemb	Semelhança com o título	55,0833	**59,1247**
AggregateSim	Similaridade agregada	43,0002	**45,0834**
TextRank	*TextRank*	38,2503	**42,1252**
BushyPath	*Bushy path*	37,0003	**38,7505**
	Média	49,2402	**53,4827**

Os resultados indicaram que é relevante ter mais de uma opção de tradução, devido a particularidade de resultados dos tradutores, enobrece-se a iniciativa da inclusão do segundo tradutor (Microsoft), que se revelou mais confiante que o seu concorrente em quase todas as abordagens de sumarização, muito em virtude de manter a estrutura original da sentença traduzida. Outro fator que justifica a utilização do método de tradução intermediária é o tempo de processamento, uma vez que os algoritmos são otimizados para a língua inglesa, dando notória agilidade em detrimento ao processamento dos demais idiomas. Por exemplo no método de nomes próprios, ele é

auxiliado pela biblioteca (CoreNLP) que contém recursos de processamento de linguagem natural para o reconhecimento de entidades nomeadas (*Name Entity Recognition*) para o inglês.

Fornecendo textos em outros idiomas supõe-se que o tempo de processamento será incrementado ou gerar erros de processamento. Diante desta suposição, executou-se mais um experimento a fim de verificar tal diferença entre o processamento utilizando o método de tradução e a execução sem sua utilização. Os experimentos foram executados seguindo os seguintes passos:

Passo 1: Gerar sumários de 4 sentenças pelos algoritmos de sumarização usando corpus espanhol, sem utilizar a tradução intermediária, contabilizando os tempos de processamento;

Passo 2: Gerar sumários de 4 sentenças pelos algoritmos de sumarização usando corpus espanhol, utilizando o método de tradução intermediária com a API Google, contabilizando os tempos de processamento;

Passo 3: Gerar sumários de 4 sentenças pelos algoritmos de sumarização usando corpus espanhol, utilizando o método de tradução intermediária com a API Microsoft, contabilizando os tempos de processamento;

Os resultados obtidos pelo experimento em questão na Tabela 17.

Tabela 17. Tempo de processamento em *s* dos sumarizadores utilizando ou não o processo de tradução.

Abreviação	Dependente de idioma?	Corpus espanhol	Traduzido Google	Traduzido Microsoft
WordFreq	Não	9,954	9,524	9,640
TF/IDF	Não	173,107	109,470	108,418
UpperCase	Não	7,612	7,780	8,128
ProperNoun	Sim	258,353	24,046	20,650
WordCo-oc	Não	18,428	19,170	18,142
LexicalSim	Sim	259,550	339,545	331,398
Cue-phrases	Sim	8,890	8,650	10,617
NumericalData	Não	7,770	7,137	12,545
SentenceLength	Não	7,902	6,822	8,266
SentencePos_1	Não	8,004	6,421	7,693
SentencePos_2	Não	7,329	6,898	9,555
SentenceCent_1	Não	10,636	10,372	13,613
SentenceCent_2	Sim	50,099	30,306	36,344
TitleResemb	Não	11,389	7,133	9,363
AggregateSim	Sim	9,528	8,504	10,484
TextRank	Sim	1107,350	151,373	231,220
BushyPath	Sim	10,798	9,906	11,240
	Média geral	**115,688**	**44,886**	**50,430**
	Média por documento	0,28922s	0,112s	0,126s

De acordo com a tabela em face, os resultados confirmaram a hipótese levantada. Em geral, além dos algoritmos que se sabe da utilização de bibliotecas ou dicionários dependentes de idioma (como por exemplo, os de nomes próprios e *cue-phrases*) os algoritmos de sumarização que não apresentaram alteração o seu tempo médio de processamento são métodos independentes de idioma, já os que tiveram aumento substancial no tempo de processamento são os métodos dependentes de idioma, justificando assim o tratamento separado de tais métodos pelo módulo de tradução intermediária.

5.1.4 Sumarização

Este módulo da plataforma disponibiliza os 17 métodos de sumarização extrativa descritos e analisados profundamente no capítulo 4. Incluindo a possibilidade de combinações destes algoritmos, as quais viabilizam o processo de sumarização combinada via média aritmética das pontuações obtidas pelos métodos selecionados, sejam eles dependentes ou independentes de idioma. Neste sentido, propõe-se um experimento utilizando um exemplo da combinação métodos de sumarização. A escolha dos métodos a serem combinados foi baseada nos resultados obtidos pelos algoritmos documentados no capítulo 4, considerando os melhores resultados e os menores tempo de processamento, sendo eles: Frequência de palavras, tamanho e posição de sentenças. O experimento em questão será detalhado mais adiante, na seção 5.2.

5.1.5 Pontuação e Seleção de Sentenças

No processamento, cada abordagem calcula os valores para as sentenças do documento, tais valores são agregados e classificados; as sentenças mais bem pontuadas são selecionadas para o sumário de acordo com o limiar provido pelo usuário, que pode ser por quantidade (por exemplo, 6 sentenças) ou por percentual (ou seja, 30% de um original com 20 corresponde a 6 sentenças).

Por fim, executa-se uma verificação para retirada de possíveis ruídos da tradução, usando um mapeamento bijetivo entre as sentenças processadas, pontuadas e selecionadas com as do texto original, consistindo na validação das sentenças contidas no sumário gerado com as frases do documento original usando o índice de cada sentença, diante do fato do índice não ser alterado pelo processamento. Para exemplificar, pode ser vista uma amostra do processo como um todo a seguir.

(a) Conteúdo original de uma notícia em espanhol.

```
[1] (CNNMéxico) — El jamaicano Usain Bolt, que consiguió este domingo su tercera medalla
de oro en Moscú, su octava medalla de oro en campeonatos del mundo y la décima en total,
se dijo orgulloso de sí mismo y anunció que seguirá trabajando "para dominar tanto tiempo
como sea posible".
[2] "Da gusto vencer", dijo, luego de ganar con el equipo jamaicano el primer lugar en la
carrera de relevos 4x100, según EFE.
[3] "Para eso he estado entrenando.
[4] He trabajado mucho y muy duro, superando todos los obstáculos que he ido encontrando
en mi camino.
[5] Estoy orgulloso de mí mismo y voy a seguir trabajando para dominar tanto tiempo como
sea posible".
[6] Bolt no tiene claros sus planes inmediatos para cerrar la temporada.
[7] "No estoy en la forma en que me gustaría estar, así que vamos a ver qué pasa con la
final de la Diamond League", dijo, según EFE.
[8] Cuestionado sobre si estaba pensando en tomarse un descanso la próxima temporada,
dijo: "No lo sé todavía, mi entrenador (Glen Mills) decidirá".
[9] El seis veces campeón olímpico enmarcó sus triunfos dentro del desarrollo del atletismo
en su país.
[10] "Los talentos de la velocidad pueden surgir en cualquier sitio, no solo en Jamaica,
pero el atletismo se está haciendo cada vez más grande en Jamaica y creo que pronto veremos
muchos otros talentos", dijo.
[11] Con la presea dorada de este domingo, Bolt se convirtió en el atleta más premiado de
la historia, superando a la leyenda estadounidense de las pistas Carl Lewis, el Hijo del
Viento.
[12] Bolt y Lewis acumulan las mismas medallas en mundiales de atletismo, 10.
[13] Sin embargo, el jamaiquino suma 8 de oro y 2 de plata, con un valor superior a las
de Lewis que acumuló 8 de oro, 1 de plata y 1 de bronce.
[14] En días anteriores, durante las actividades del Mundial de Atletismo en Moscú, Rusia,
Bolt recuperó su corona en los 100 metros y también se posicionó como el mejor en los 200.
[15] La última victoria de Bolt, al igual que la de sus compatriotas, llega como aire
fresco para el deporte en Jamaica, sacudido en los últimos meses por escándalos de dopaje
como el del velocista Asafa Powell.
```

(b) Sumário gerado pelo Microsoft Translator API.

```
[1] (CNNMéxico) - the Jamaican Usain Bolt, who won on Sunday their third gold medal in
Moscow, its eighth gold medal in the World Championships and tenth overall, said proud of
itself and announced that it will continue to work "to dominate as long as possible".
[2] "It gives taste to beat," he said, after winning first place with the Jamaican team
in the 4 x 100, according to EFE relay race.
[11] With the gold medal on Sunday, Bolt became the most awarded athlete of history,
beating American tracks legend Carl Lewis, the son of the wind.
[13] However, the Jamaican added 8 gold and 2 silver, with a value greater than the Lewis
who amassed 8 gold, 1 silver and 1 bronze.
[14] In earlier days, during the activities of the Athletics World Cup in Moscow, Russia,
Bolt regained his Crown in the 100 meters and also ranked as the best in the 200.
[15] The last victory of Bolt, as well as of their compatriots, arrives as fresh air for
the sport in Jamaica, shaken in recent months by scandals of doping as the sprinter Asafa
Powell.
```

(c) Sumário gerado pelo Google Translate API.

```
[1] (CNNMéxico) - Jamaican Usain Bolt, who won his third Sunday gold medal in Moscow, his
eighth gold medal at the world championships and tenth in total, said proud of himself
and announced that he will continue to work "To dominate as long as possible."
[2] "It's nice to win," he said after winning the Jamaican team first place in the 4x100
relay race, according to EFE.
[11] With the gold medal on Sunday, Bolt became the athlete's awarded in history, beating
the American legend of the tracks Carl Lewis, the Son of Wind.
[13] Without But the Jamaican sum 8 gold and 2 silver, with a higher value to Lewis
accumulated 8 gold, 1 silver and 1 bronze.
[14] In earlier days, during the Global Athletics in Moscow Russia, Bolt regained his
crown in the 100 meters and also ranked as the best in the 200.
[15] Bolt's last victory, like that of his countrymen, comes as air cool for the sport in
Jamaica, rocked in recent months by scandals doping as the sprinter Asafa Powell.
```

Nota-se o quão importante é o processo de tradução, mesmo após a tarefa, os índices das sentenças são mantidos, o que permite o mapeamento bijetivo, mesmo que haja ruídos, nenhuma alteração é realizada nos índices, permitindo o mapeamento seguro a documento original, justificando, assim, a validade da estratégia de independência.

(d) Sumário final do processo após o mapeamento bijetivo.

```
[1] (CNNMéxico) — El jamaicano Usain Bolt, que consiguió este domingo su tercera medalla
de oro en Moscú, su octava medalla de oro en campeonatos del mundo y la décima en total,
se dijo orgulloso de sí mismo y anunció que seguirá trabajando "para dominar tanto tiempo
como sea posible".
[2] "Da gusto vencer", dijo, luego de ganar con el equipo jamaicano el primer lugar en la
carrera de relevos 4x100, según EFE.
[11] Con la presea dorada de este domingo, Bolt se convirtió en el atleta más premiado de
la historia, superando a la leyenda estadounidense de las pistas Carl Lewis, el Hijo del
Viento.
[13] Sin embargo, el jamaiquino suma 8 de oro y 2 de plata, con un valor superior a las
de Lewis que acumuló 8 de oro, 1 de plata y 1 de bronce.
[14] En días anteriores, durante las actividades del Mundial de Atletismo en Moscú, Rusia,
Bolt recuperó su corona en los 100 metros y también se posicionó como el mejor en los 200.
[15] La última victoria de Bolt, al igual que la de sus compatriotas, llega como aire
fresco para el deporte en Jamaica, sacudido en los últimos meses por escándalos de dopaje
como el del velocista Asafa Powell.
```

De acordo com o exemplo, o mapeamento bijetivo revela-se importante, abrindo um leque de utilidades, por exemplo, na sumarização de resultados de pesquisa na web, resumindo uma quantidade significativa de informações que devem ser analisadas em um curto espaço de tempo, entre muitas outras utilidades. O recurso multilíngue foi um ponto chave para obter sumários a partir de diferentes idiomas eliminando os ruídos da tradução.

5.2 Experimentos e Resultados

Nesta seção apresentam-se os resultados experimentais da metodologia e uma análise para avaliar a qualidade dos resumos gerados pela plataforma. Os experimentos utilizaram três diferentes corpora, com idiomas distintos: CNN em inglês, CNN em espanhol e TeMário em português. Os corpora detêm as seguintes características: notícias que contêm apenas texto; sumário de alta qualidade provido por profissionais; e um resumo padrão-ouro gerados por seres humanos. O ROUGE (*Understudy Lembre-Oriented para Gisting Avaliação*) (LIN, 2004) foi utilizado para avaliar a qualidade dos sumários gerados pela plataforma. É um método quantitativo, baseado em estatísticas de n-gramas e é altamente correlacionado com as avaliações humanas (LIN & HOVY, 2003). Este avaliador totalmente automatizado essencialmente afere a similaridade de conteúdo entre sumários gerados pelo sistema e os respectivos resumos padrão-ouro (gerados por seres humanos). A avaliação é realizada usando a configuração n-gramas (1,1) do ROUGE, na qual apresenta a maior correlação com julgamentos humanos a um nível de confiança de 95%. A seguir, apresentam-se experimentos separados por corpus.

5.2.1 CNN em Inglês

Conforme já apresentado no capítulo anterior, este *corpus* desenvolvido por Lins e seus colaboradores (LINS, *et al.*, 2012) contém artigos de notícias extraídas do site da CNN americana. A versão de testes deste corpus apresenta 400 textos atribuídos a 10 categorias: Ásia, negócios, Europa, América Latina, Oriente Médio, EUA, esportes, tecnologia, viagens e notícias do mundo. Os resultados do cálculo ROUGE para cada um dos algoritmos, utilizando o corpus em destaque, usando como sumário padrão-ouro os sumários gerados por humanos, podem ser visualizados na tabela abaixo. Foram gerados sumários contento 6 sentenças por todos os métodos analisados, visando similaridade de experimentos com os executados no capítulo 4.

Tabela 18. Resultados do ROUGE-1 para o *dataset* CNN em inglês [1].

Abreviação	Average_R	Average_P	Average_F
Plataforma	**0,71 (±0,24)**	0,29 (±0,13)	0,41 (±0,16)
WordFreq	**0,71 (±0,19)**	0,35 (±0,13)	0,46 (±0,15)
SentenceLength	0,70 (±0,18)	0,33 (±0,12)	0,44 (±0,15)
SentencePos_1	0,61 (±0,22)	**0,40 (±0,13)**	**0,47 (±0,15)**

O resultado exibido é mostrado com o valor da medida associada à coluna (Cobertura, Precisão ou *F-measure*) e entre parênteses, encontra-se o desvio-padrão registrado pelo ROUGE-1. Embora haja proximidade de valores nos resultados, alguns pontos merecem destaque.

- A plataforma manteve a alta Cobertura (0,71), semelhantemente ao método WordFreq (frequência de palavras);
- SentencePos_1 (posição de sentença) tem as melhores taxas médias de precisão e *F-measure*;
- A plataforma proposta apresentou valores entre 3 e 6 pontos percentuais em termos de medida combinada *F-measure* abaixo dos algoritmos de sumarização analisados individualmente.

Neste experimento a plataforma combinou os resultados usando uma média aritmética para as pontuações de cada método utilizado internamente, visando melhores resultados, outro experimento foi efetuando utilizando uma média ponderada das pontuações obtidas por cada método utilizado, obtendo-se melhoria nos resultados.

Tabela 19. Resultados do ROUGE-1 para o *dataset* CNN em inglês [2].

Abreviação	Average_R	Average_P	Average_F
Plataforma	0,66 (±0,18)	**0,38 (±0,14)**	**0,47 (±0,15)**

De acordo com os resultados obtidos, a plataforma proposta conseguiu o melhor resultado médio para *F-measure* (0,47), de forma semelhante à estratégia SentencePos_1 (posição de sentença), segunda melhor precisão média (ficando apenas 0,02 pontos abaixo da concorrente SentencePos_1).

Como no capítulo 4, optou-se ainda por efetuar-se a avaliação híbrida descrita na seção 4.2.3.2, uma vez que o ROUGE-1 é meramente quantitativo. Nela as pontuações obtidas pelos algoritmos indicam quantas sentenças dos sumários gerados por humanos (padrão-ouro) foram escolhidas pelos métodos avaliados. A figura a seguir apresenta os resultados desta avaliação híbrida.

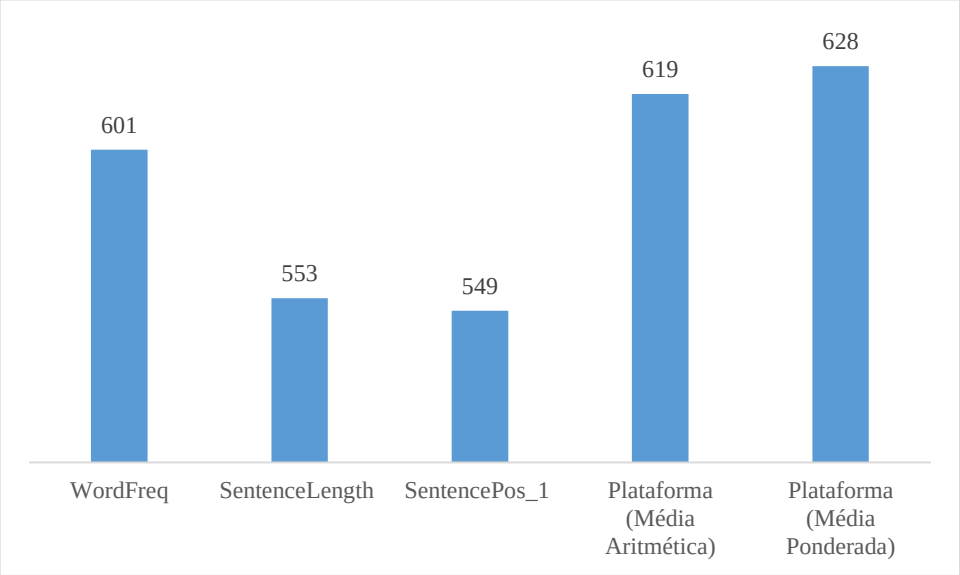

Figura 11. Número de sentenças corretas dos métodos implementados, segundo a avaliação híbrida.

Observa-se a plataforma supera a concorrência, ou seja, os métodos combinados são melhores que individualmente, devido à média aproveitar o melhor de cada algoritmo selecionado para combinação, dando maior importância às sentenças que obtiveram maior pontuação, em detrimento das que obtiveram menores pontuações. Além disso, registraram-se os tempos de processamento de cada um, apresentando-os a seguir.

Tabela 20. Tempo de execução utilizando 400 textos do corpus CNN.

Abreviação	**Tempo de execução *(s)***
Plataforma	13,251
WordFreq	12,895
SentenceLength	9,030
SentencePos_1	11,352

Apesar da plataforma utilizar os 3 métodos combinados, o seu tempo de processamento não é muito diferente dos métodos individuais, devido o maior peso na

computação do tempo deve-se ao fato do acesso ao disco (leitura e gravação), em termos de complexidade computacional, os métodos são aritmeticamente similares.

Ao longo do trabalho um novo corpus, foi desenvolvido, utilizando a captação automática do site CNN.com, e utilizando um processo de mapeamento distribuído, onde se tinha uma dupla de especialistas mapeando um grupo de notícias, tendo-se os mapeamentos revisados por um terceiro especialista, mantendo-se as concordâncias e reavaliando-se as divergências, dando maior imparcialidade ao mapeamento e ao processo de construção como um todo.

Tal corpus encontra-se com mais de 3000 notícias com sumários gerados por humanos, para fins de demonstração dos resultados anteriores, executa-se um experimento com uma porção de 1010 documentos, contendo cada um, um sumário de 4 sentenças como padrão-ouro. Deste modo, geraram-se resumos contendo 4 sentenças pela plataforma, a fim de, visualizar se haverá grandes diferenças em relação à avaliação ROUGE, frente às demais análises já demonstradas.

Tabela 21. Resultados do ROUGE-1 para o *dataset* CNN em inglês (Combinação ponderada).

Abreviação	Average_R	Average_P	Average_F
Plataforma	0,56 (±0,17)	0,45 (±0,16)	0,49 (±0,15)
WordFreq	0,57 (±0,17)	0,43 (±0,17)	0,48 (±0,16)
SentenceLength	0,52 (±0,16)	0,38 (±0,15)	0,43 (±0,16)
SentencePos_1	0,42 (±0,19)	0,45 (±0,17)	0,43 (±0,17)

Como é possível observar, os resultados mantêm-se satisfatórios, a melhoria da precisão advém da diminuição de sentenças contida nos sumários gerados, no experimento anterior, tinha-se gerado 6 sentenças por sumário, aumentando a cobertura mas sofrendo penalidades quanto à precisão, neste último experimento, gerou-se 4 sentenças por sumário, aumentando a precisão penalizando um pouco à cobertura, mas obtendo-se uma medida combinada (*F-measure*) ainda melhor que os experimentos anteriores e aos demais concorrentes. Pode-se afirmar que independente do corpus utilizado, os sumários gerados pela plataforma podem ser classificados como resumos competitivos ou de boa qualidade.

5.2.2 CNN em Espanhol

Este corpus foi desenvolvido neste trabalho seguindo o mesmo método de criação do *corpus* CNN em inglês (LINS, *et al.*, 2012). Contém artigos de notícias extraídas do site CNN México. A versão de testes deste corpus apresenta 400 textos atribuídos a 08 categorias: esportes, entretenimento, mundo, nacional, opinião, tecnologia, viagem e

notícias de saúde. Nesta versão do experimento, utilizou-se a plataforma de sumarização sobre o *corpus* em referência usando como sumário padrão-ouro os *highlights* disponíveis no site CNN México. Os resultados experimentais fornecidos pelo ROUGE para o *dataset* em referência são apresentados na Tabela 22. Uma vez que este corpus é introduzido aqui, então não há resultados de outros métodos de sumarização a serem analisados.

Tabela 22. Resultados ROUGE-1 para o *dataset* CNN em espanhol usando *highligths* como padrão-ouro.

Abreviação	Average_R	Average_P	Average_F
Plataforma	**0,72 (±0,11)**	0,11 (±0,03)	0,20 (±0,05)

A cobertura média da plataforma está próxima dos valores obtidos pela mesma em outros corpora, apresentados nas Seções 5.2.1 e 5.2.3. Isso pode indicar que a quantidade de erros introduzido por intermédio do passo de tradução não afetou o processo global. A baixa precisão dá-se pelo fato dos sumários padrão-ouro utilizados (*highlights*) não coincidirem com as sentenças dos sumários obtidos ou do texto original, tratam-se de sumários abstrativos e não extrativos, como por exemplo, os sumários sugeridos por humanos.

Deste modo, providenciou-se uma nova versão do *corpus*, utilizando-se a mesma metodologia utilizada na construção do *corpus* atual da versão Inglesa, com sumários sendo mapeados por duplas de especialistas, tendo-se um terceiro como revisor da anotação, mantendo-se as concordâncias e reavaliando-se as divergências. Tal corpus encontra-se com mais de 1000 notícias com sumários gerados por humanos, e para fins de demonstração dos resultados anteriores, executa-se um experimento com uma porção de 510 documentos, contendo cada um, um sumário padrão-ouro contendo entre 2 e 4 sentenças. Assim, geraram-se resumos contendo 4 sentenças pela plataforma, a fim de, comprovar a diferenças entre a avaliação ROUGE-1 usando a primeira versão do *corpus*, que utilizou os *highlights* como padrão-ouro, e a validação usando a versão mais atual do *corpus* que possui os sumários gerados por humanos como resumo ideal.

Tabela 23. Resultados ROUGE-1 para o *dataset* CNN em espanhol usando *sumários gerados por humanos* como padrão-ouro.

Abreviação	Average_R	Average_P	Average_F
Plataforma	0,46 (±0,18)	0,39 (±0,19)	0,42 (±0,13)

Conforme se previu, a melhoria é substancial utilizando sumários gerados por humanos, ou seja, o esforço administrativo dispendido na anotação é uma tarefa essencial para construção de um corpus de qualidade. A plataforma manteve os resultados

competitivos, compatíveis com os obtidos nos experimentos em língua inglesa, corroborando sua qualidade na geração dos sumários, independentemente do idioma utilizado, visando uma validação ainda maior, a seguir, tem-se mais um experimento, agora utilizando um *corpus* terceiro em língua portuguesa, trata-se do corpus TeMário.

5.2.3 TeMário em Português

Este *corpus* é uma coleção de teste (PARDO & RINO, 2003) que contém 100 artigos de notícias dos jornais brasileiros: Jornal de Brasil e Folha de São Paulo. Os documentos foram selecionados para cobrir uma variedade de domínios (por exemplo, mundo, política, negócios, editoriais) e um especialista em português brasileiro produziu sumários de forma manual para criação do padrão-ouro.

Os resultados experimentais fornecidos pelo ROUGE para o *dataset* em questão são mostrados na Tabela 24. Para fins de comparação, alguns dos resultados experimentais obtidos por Leite e Rino (LEITE & RINO, 2008) foram utilizados aqui. Eles usaram o mesmo corpus em seus experimentos com o objetivo de avaliar a combinação de vários recursos de aprendizagem de máquina para a sumarização automática. Eles utilizaram a medida ROUGE-1 com 30% de taxa de sumarização, os resumos manuais foram usados como padrão-ouro. No entanto, apenas a cobertura média é registrada no trabalho deles. Para uma comparação idônea, utilizaram-se nos experimentos as mesmas configurações.

Tabela 24. Resultados do ROUGE para o *dataset* TeMário em português.

Abreviação	**Average_R**
Plataforma	**0,77**
SuPor2-LR*	0,53
SuPor-2*	0,52
TextRank*	0,51
BestCN*	0,50
Baseline *	0,49

* (LEITE & RINO, 2008)

Neste experimento a plataforma combinou os resultados usando uma média aritmética para as pontuações de cada método utilizado internamente, já alcançando o melhor resultado entre os concorrentes, como o trabalho citado apenas apresenta o resultado de Cobertura média, só poderíamos comparar utilizando o mesmo dado, entretanto assim como nos experimentos anteriores, outro experimento foi efetuado utilizando uma média ponderada das pontuações obtidas por cada método utilizado, sendo apresentado abaixo com o resultado completo (Cobertura, Precisão e *F-measure* médios).

Tabela 25. Resultados do ROUGE para o *dataset* TeMário em português.

Abreviação	Average_R	Average_P	Average_F
Plataforma	0,65 (±0,12)	0,62 (±0,15)	0,62 (±0,11)

Conforme se pode observar, tem-se outro resultado competitivo usando desta vez um *corpus* em língua portuguesa. O alto acerto se explica pelo fato dos textos possuírem períodos de tamanho médio a longo, facilitando a pontuação das sentenças e suas respectivas escolhas. Levando-se em consideração todas as funcionalidades utilizadas na plataforma, seu processo de construção, os resultados parciais e finais obtidos, a proposta de utilizar um método de sumarização independente, envolvendo aspectos de identificação, tradução e sumarização revelam-se relevante, incluindo os baixos tempos de processamentos registrados, denotando ampla aplicabilidade da proposta, incluindo-se a possibilidade futura de sua utilização na web e até em sistemas embarcados em dispositivos móveis.

5.3 Considerações Finais

Neste capítulo, apresentou-se a Sumarização Independente de Idiomas em detalhes, incluindo sua arquitetura geral, a integração dos módulos descritos nos capítulos anteriores, além do detalhamento de outros módulos importantes tais como o de tradução automática intermediária. Os experimentos e resultados para diferentes idiomas ajudam a dar a noção de independência defendida nesta tese. A seguir apresentam-se as conclusões em torno da tese e os trabalhos futuros possíveis de serem realizados visando a melhoria da plataforma proposta.

6 Conclusões e Trabalhos Futuros

6

Você faz suas escolhas e suas escolhas fazem você.
William Shakespeare

A mente que se abre a uma nova ideia, jamais voltará ao seu tamanho original.
Albert Einstein

6.1 Considerações e Oportunidades de Trabalhos Futuros

Em virtude dos fatos apresentados, a plataforma de sumarização independente de idioma visa à criação de pequenas versões dos documentos visando ajudar na análise de conteúdo multilíngue, seja na web ou em outra aplicação à qual a mesma seja aproveitada. Uma arquitetura usando serviços integrados baseados na identificação de idiomas, tradução automática e sumarização, onde cada método foi escolhido através de estudos e experimentos registrados ao longo dos capítulos contidos neste documento, observando sempre os melhores resultados e os menores tempos de processamento. Foram utilizados Três diferentes corpora com idiomas distintos visando testar e avaliar a plataforma.

As principais contribuições deste trabalho foram: (a) a plataforma de sumarização independente de idioma; (b) com suporte para até 25 idiomas diferentes com o processo intermediário de tradução e um método combinado a sumarização; (c) as avaliações mostram resultados compatíveis se comparados a outros trabalhos recentes de propósito semelhante. Além disso, a plataforma de sumarização é facilmente extensível e com diferencial da adição de novas linguagens ou novos métodos de sumarização ser simples e objetiva, precisando apenas o desenvolvimento do método a ser adicionado.

Estudos para melhoria da plataforma são salutares em um futuro próximo, visando aumentar o suporte a outros idiomas e adição métodos de sumarização para produzir resultados ainda melhores, inclusive focando em aspectos abstrativos e funcionais de sumarização. Podendo-se ainda averiguar a possibilidade do fornecimento do sumário de saída no idioma desejado pelo usuário, permitindo que um texto em um idioma qualquer seja sumarizado e que o resumo obtido seja mostrado no idioma de preferência do usuário.

Tal possibilidade pode ser alcançada pela utilização do próprio módulo de tradução intermediária, entretanto, diferentes métodos de eliminação de ruídos devem ser estudados uma vez que dependendo da escolha do usuário, o texto original poderá passar por 2 traduções, sem a possibilidade da utilização do mapeamento bijetivo.

As estratégias utilizadas na fase de Pontuação e Seleção de sentenças podem ser melhoradas e ajustadas, mas dependem de obtenção de um *corpus* de teste maior. Os esforços nessa direção já foram iniciados e estão em andamento.

Em adendo, as mais importantes estratégias de sumarização encontradas na literatura nos últimos 10 anos foram registradas, implementadas, experimentadas e avaliadas. Dentre as melhores, segundo a avaliação híbrida, registram-se as estratégias de Frequência de Palavras (*WordFreq*), TF/IDF (TF/IDF), Similaridade léxica (*LexicalSim*), Tamanho e posição de sentenças (*SentenceLength* e *SentencePos*_1) além da Semelhança com o título (*TitleResemb*). Os resultados fornecidos pelo ROUGE na avaliação quantitativa dos métodos de sumarização foram bem próximos aos obtidos pela análise híbrida. O TF/IDF é de longe o que mais consome recursos computacionais entre todos os métodos testados (TF/IDF). Os métodos de Frequência de palavras (WordFreq) e Tamanho da sentença (*SentenceLength*) proporcionaram o melhor equilíbrio no desempenho entre tempo de execução e eleição de sentenças relevantes. Apesar de já apresentarmos um exemplo de combinação neste trabalho, é salutar futuramente um estudo aprofundado das estratégias para melhor combinação dos resultados obtidos, visando obter ainda melhores resultados.

A criação dos *corpora* CNN em inglês e espanhol não podem ser deixados de lado, pois trata-se de um *corpus* de alta qualidade, construído visando avaliações utilizando o mesmo *dataset*, hardware e linguagem de programação. Dentre as avaliações registradas, tem-se a da estratégia para a sumarização de texto tomando várias ferramentas de sumarização como entrada, conseguindo compor os resultados para produzir um resumo melhor. Além da avaliação sistemática de seis ferramentas de sumarização disponíveis. Na avaliação quantitativa feita usando ROUGE foi identificado o *TextCompactor* como o melhor das seis ferramentas *blackbox* testadas, bem como a estratégia de sumarização composta que obteve a segunda melhor colocação.

Ainda considerando os resultados obtidos pela avaliação de identificadores de idiomas, também de forma imparcial, sob o mesmo corpus, hardware e linguagem de programação, que define uma comparação justa entre os envolvidos. Em adendo, propôs-se um novo algoritmo híbrido de identificação de idiomas, denominado CALIM. Que por

sua vez, ao lado do *LangDetector*, mostraram um desempenho competitivo no experimento usando o *Europarl test corpus*, não só devido às maiores precisões, mas também pelos tempos de processamento. O primeiro foi mais rápido, o segundo foi mais preciso, mas com uma pequena margem de ganho. No experimento com o *Europarl full corpus*, os algoritmos acima mencionados aparecem com os melhores resultados em termos de precisão, com praticamente a mesma pontuação desempenho neste segundo *dataset*, sendo o CALIM mais rápido do que LangDetect, entendendo-se como melhor algoritmo de identificação de idiomas envolvendo custo-benefício.

Assim como nos processos anteriores, também é válida em um futuro próximo estudos aprofundados visando melhorias neste módulo, visando não só o aumento do número de idiomas suportados, mas também a otimização das condições de escolha do algoritmo, visando obter maior suporte e melhor performance.

6.2 Contribuições

Dentre as contribuições alcançadas neste trabalho, podem-se enumerar:

4. Desenvolvimento de uma plataforma para sumarização independente de idiomas, contendo:
    a. Um módulo de identificação de idiomas, retornando o idioma do documento original com bom custo benefício. Tal método de identificação foi inovador por combinar de técnicas de reconhecido sucesso na literatura, que geralmente eram usadas separadamente, obtendo-se como resultado uma identificação rápida e precisa, evidenciando nos experimentos o melhor custo benefício;
    b. Um módulo de tradução que efetua a tradução do texto original para a língua inglesa, requerida por alguns dos algoritmos de sumarização que necessitam de algum processo dependente de idioma, requer como entrada o idioma de origem e destino, além do conteúdo a ser traduzido;
    c. Um módulo de sumarização que reúne 17 diferentes técnicas de sumarização, agrupadas em 3 grupos, que podem ser utilizadas individualmente ou combinadas a critério do usuário. Algumas são independentes de língua, tais como as técnicas baseadas em estatística como TF/IDF, já outras são dependentes de língua, como *CuePhrases*, que usa um dicionário de palavras que sinalizam sentenças importantes

no texto, para essas técnicas, os fluxos de identificação e tradução são utilizados, para as demais, a sumarização é efetuada de maneira direta;

d. A saída por padrão é fornecida no idioma do documento original, onde preventivamente usa-se um mapeamento bijetivo para assegurar que as sentenças do sumário gerado estejam idênticas às sentenças do texto original. Como existe um módulo tradutor integrado, há possibilidade de exibir o sumário gerado em um idioma especificado pelo usuário.

5. Para os experimentos foi necessário:
    a. A criação de um corpus de fácil entendimento e leitura, escolhendo-se notícias do portal CNN, tal corpus foi coletado, processado, revisado e obtido sumários abstrativos sugeridos pelos autores, chamados de *highlights*, chegando a 2000 documentos e em contínuo crescimento; Criaram-se sumários extrativos de referência (*gold standard*), escolhidos por um grupo de especialistas após analisarem uma porção das notícias presentes no corpus (cerca de 400), os quais foram classificados como sumários de confiança para fins de avaliação;
    b. Na avaliação, tais sumários de qualidade *gold standard*, são utilizados para averiguar quais das técnicas de sumarização selecionadas conseguem coincidir suas sentenças de saída, com as contidas no *gold standard*, este resultado será proporcional ao número de coincidências. Tal método possui aspectos quantitativos e qualitativos (híbrido) que não são encontrados nos demais métodos de avaliação de sumários.
    c. Sumarização extrativa automática de textos utilizando combinação de métodos obtendo resultados relevantes frente aos demais sistemas analisados;
    d. Utilizou-se ainda a medida de avaliação consagrada pelos demais trabalhos da área, ROUGE, visando reforçar os resultados obtidos pelo método proposto;
    e. Visando cumprir o experimento multilíngue, além do corpus em língua inglesa, construiu-se um corpus adicional em espanhol, baseado nas notícias CNN México, que também possuem sumários gerados pelos autores, além de sumários extrativos gerados por especialistas. Também como parte do experimento multilíngue foi usado o corpus em Português do Brasil, chamado TeMário (PARDO & RINO, 2003)

no qual os textos são originalmente de notícias de jornais do Brasil, contendo sumários gerados automaticamente e gerados por especialistas.

6. Em particular, os experimentos nos levaram aos seguintes resultados relevantes:
   - Resultados relativos à identificação de idiomas indicaram que o melhor custo benefício foi obtido pelo método aqui proposto chamado CALIM, apresentando resultados de alta precisão e baixo tempo de processamento:
     - 97.08% de precisão em 10s de processamento para 21 mil documentos *Plain Text* com tamanho médio de 172.90 bytes;
     - 99.99% de precisão em 2,776s de processamento para cerca de 60 mil documentos XML de tamanho médio de 84.51 KB.
   - Resultados relativos ao módulo de tradução, a Microsoft API mostrou-se melhor que o Google, mantendo maior taxa de confiança de escolha de sentenças pelos sumarizadores após o processo de tradução, com eficiência 4% superior frente ao segundo colocado;
   - Dentre os experimentos entre as técnicas de sumarização, as melhores foram às baseadas em palavras e sentenças, por exemplo, Pontuação de Palavra, Frequência de Termos, além de Posicionamento e Tamanho de Sentenças, utilizando o método de avaliação híbrido proposto.
   - No caso do experimento multilíngue, a plataforma integrada igualou (ROUGE) e superou (Avaliação híbrida) os resultados frente aos métodos monolíngues (testados apenas em idioma inglês), além de superar trabalhos relacionados usando o mesmo *corpus* e configuração ROUGE (TeMário), ainda provendo resultados no *corpus* espanhol, como o corpus foi criado neste trabalho, e por não ser de domínio público, não existem resultados de outros métodos sobre tal corpus.
   - Por fim, além da plataforma de sumarização multilíngue, têm-se as contribuições da criação dos maiores corpora para testes de sumarização já registrados (CNN Inglês e Espanhol), além de ser um trabalho multilíngue que se pôs à prova com três diferentes idiomas presentes em seus experimentos com resultados superiores aos concorrentes.

6.3 Aspectos de Pesquisa da Área Vindouros

Em virtude do que foi mencionado, diversas questões de pesquisas foram elucidadas, assim como um bom número de contribuições foram registradas ao longo do trabalho, permitindo deste modo percebermos a notória importância da área de pesquisa, a relevância do trabalho, assim como as lacunas existentes ainda proporcionarão anos e anos de pesquisa, incluindo as questões de sumarização abstrativa e funcional, pouquíssimo exploradas pela comunidade diante da sua complexidade, tendo-se uma oportunidade latente de exploração destas áreas nos trabalhos futuros.

Com o crescimento do mercado de computação móvel, tem-se outra oportunidade em abertos, devido a gama de conteúdos a que estes dispositivos são expostos cotidianamente, uma plataforma para sumarização automática de textos independente de idioma, agora na versão *mobile*, seria de grande utilidade e aplicabilidade, além do desenvolvimento desta versão da plataforma envolver desafios de pesquisa que residem na criação de um software deste nível de utilidade, com utilização de recursos de hardware mínimos, necessitando uma otimização de código e preterindo o uso de bibliotecas.

É possível ainda neste contexto, a utilização da sumarização independente de idioma visando auxiliar o aprendizado de uma segunda língua por discentes, principalmente em análise textual, não apenas a determinação de atores e ações presentes no texto, mas também a identificação do núcleo textual, em outras palavras a principal lição que a produção deseja revelar.

Por fim, é possível ainda utilizar-se de recursos da Web 2.0 como Ontologias de conhecimento geral, permitindo-se deste modo a adição de relações semânticas entre as sentenças, tendo-se revelada características adicionais não presentes no texto. Para exemplificar, pode-se imaginar a frase “Obama critica primeiro ministro russo por sua declaração.”, nesta frase, poder-se-ia obter a informação temporal (época da publicação), atores (quem é Obama? quem é o primeiro ministro Russo?) e ações (qual foi a declaração do primeiro ministro?). Tal recurso poderia auxiliar na geração textual de sumários abstrativos, visando à criação de textos ricos tanto em vocabulário quanto em informação.

Referências

ABUOBIEDA, A., SALIM, N., ALBAHAM, A. T., OSMAN, A. H., & KUMAR, Y. J. *Text summarization features selection method using pseudo genetic-based model.* International conference on information retrieval knowledge management, pp. 193-197, 2012.

AMINE, A., ELBERRICHI, Z., & SIMONET, M *Automatic language identification: an alternative unsupervised approach using a new hybrid algorithm.* International Journal of Computer Science and Applications, 7(1), 94-107, 2010.

ASLAM, J. A., & FROST, M. *An information-theoretic measure for document similarity.* Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 449-450). New York, NY, USA: ACM. doi:10.1145/860435.860545, 2003.

BAEZA-YATES, R., & RIBEIRO-NETO, B. *Modern information retrieval.* Addison Wesley, 1999.

BARRERA, A., & VERMA, R. *Combining syntax and semantics for automatic extractive single-document summarization.* Proceedings of the 13th international conference on computational linguistics and intelligent text processing, 366-377, 2012.

BARZILAY, R., & ELHADAD, M. *Using lexical chains for text summarization.* Proceedings of the ACL workshop on intelligent scalable text summarization, pp. 10-17, 1997.

BEESLEY, K. R. Language identifier: *A computer program for automatic natural-language identification of online text.* Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association (pp. 47-54). Seattle: American Translators Association, 1988.

BHARGAVA, A., & KONDRAK, G. *Language identification of names with SVMs* Human Language Technologies. The 2010 Annual Conference of the North American Chapter of the ACL, pp. 693-696. Los Angeles, CA, USA: Association for Computational Linguistics, 2010.

BOTHA, G., & BARNARD, E. *Factors that affect the accuracy of text-based language identification.* Computer Speech & Language, 26(5), 307-320, 2012.

BRIN, S., & PAGE, L. *The anatomy of a large-scale hypertextual web search engine.* 7th WWW Conference, pp. 107-117, 1998.

CABRAL, L., LINS, R., LIMA, R., & SIMSKE, S. *A Comparative Assessment of Language Identification Approaches in Textual Documents.* Proceedings of IADIS International Conference Applied Computing 2012, pp. 67-74. Madrid: IADIS, 2012.

CABRAL, L., LINS, R., LIMA, R., & SIMSKE, S. *A Hybrid Algorithm for Automatic Language Detection on Web and Text Documents.* Booklet of 11th IAPR International Workshop on Document Analysis Systems. Tours - Loire Valley, France, 2014.

CAVNAR, W. B., & TRENKLE, J. M. *N-Gram Based Text Categorization.* Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161-169, 1994.

CHANG, J., & LIN, C. *Recurrent-Neural-Network for Language Detection on Twitter Code-Switching Corpus.* Neural and Evolutionary Computing, 2014.

Context Discovery Inc. *WebSummarizer*. Acesso em: 14 de Fevereiro de 2014, Disponível em: http://www.websummarizer.com/, 2012.

CRUZ, C., & URREA, A. *Extractive Summarization Based on Word Information and Sentence Position.* Em: A. Gelbukh, Computational Linguistics and Intelligent Text Processing (Lecture Notes in Computer Science ed., Vol. 3406, pp. 653-656). Springer Berlin Heidelberg, 2005.

CUNNINGHAM, H., MAYNARD, D., BONTCHEVA, K., TABLAN, V., ASWANI, N. R., GORRELL, G., . . . PETERS, W. *Developing Language Processing Components with GATE Version 6 (a User guide).* Sheffield, UK: Department of Computer Science, University of Sheffield, 2011.

DANARSHEK, M. *Gauging similarity with n-grams:* Language independent categorization of text. Science, 843-848, 1995.

DEERWESTER, S., DUMAIS, S., FURNAS, G., LANDAUER, T., & HARSHMAN, R. *Indexing by latent semantic analysis.* Journal of the American Society for Information Science, 41(6), 391-407, 1990.

DUNNING, T. *Statistical identification of language.* New Mexico: Computer Research Lab, New Mexico University, 1994.

EDMUNDSON, H. P. *New methods in automatic extracting.* Journal ACM, 16(2), 264-285, 1969.

ELMAANI, A. *Smmry*. Acesso em: 12 de Fevereiro de 2014, Disponível em: http://smmry.com/, 2009.

ERKAN, G., & RADEV, D. *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization.* J Artif Intell Res (JAIR), 22:457-479, 2004.

EVANS, D. K., MCKEOWN, K., & KLAVANS, J. L. *Similarity-based Multilingual Multi-Document Summarization.* IEEE Transactions on Information Theory, 49, 2005.

FATTAH, M., & REN, F. *Ga, mr, ffnn, pnn and gmm based models for automatic text summarization.* Computer Speech and Language, 23(1), 126-144, 2009.

FERREIRA, R., CABRAL, L., LINS, R., SILVA, G., FREITAS, F., CAVALCANTI, G., . . . FAVARO, L. *Assesing sentence scoring techniques for extrative text summarization.* Expert Systems with Applications, 5755-5764, 2013.

_____. *Free Summarizer*. Acesso em: 12 de Fevereiro de 2014, Disponível em: http://freesummarizer.com/, 2014

GEIGER, W., RAUCH, J., & MAIR, P. *Text Categorization in R: A Reduced N-gram* Approach. Em: W. A. Gaul, A. Geyer-Schulz, L. Schmidt-Thieme, & J. Kunze, Challenges at the Interface of Data Analysis, Computer Science, and Optimization (pp. 341-349). Karlsruhe: Springer, 2012.

GLICKMAN, O. *Applied textual entailment: A generic framework to capture shallow semantic inference.* VDM Verlag, 2009.

GOLD, E. M. *Language identification in the limit.* Information and Control, 447-474, 1967.

GOOGLE. *Google Translate API.* (Google Developers) Acesso em: 11 de Março de 2014, Disponível em: https://developers.google.com/translate/?hl=pt-BR, 2012.

GUPTA, P., PENDLURI, V. S., & VATS, I. *Summarizing text by ranking text units according to shallow linguistic features.* 13th International conference on advanced communication technology, pp. 1620-1625, 2011.

GUPTA, V. *Hybrid Algorithm for Multilingual Summarization of Hindi and Punjabi Documents.* Lecture Notes in Computer Science. Mining Intelligence and Knowledge Exploration, 8284, 717-727, 2013.

HACHEY, B., MURRAY, G., & REITTER, D. *Dimensionality reduction aids term co-occurrence based multi-document summarization.* Proceedings of the workshop on task-focused summarization and question answering, 1-7, 2006.

HAQUE, R., NASKAR, S., WAY, A., COSTA-JUSSA, M., & BANCHS, R. *Sentence similarity-based source context modelling in pbsmt.* Proceedings of the 2010 international conference on asian language processing, 257-260, 2010.

HUGHES, B., BALDWIN, T., BIRD, S., NICHOLSON, J., & MACKINLAY, A. *Reconsidering language identification for written language resources.* Proceedings of LREC2006, pp. 485-488, 2006.

INGLE, N. C. *A Language Identification Table.* The Incorporated Linguist 15(4), 98-101, 1976.

Intellexer Inc. *Intellexer Summarizer.* Acesso em: 14 de Fevereiro de 2014, Disponível em: http://summarizer.intellexer.com/, 2012.

KAMPER, H., & NIESLER, T. *A literature review of language, dialect and accent identification.* Technical Report SU-EE-1201. Digital Signal Processing Laboratory. Department of Electrical and Electronic Engineering. South Africa: Stellenbosch University, 2012.

KIKUI, G.-I., & YOKOSUKA-SH, T. *Identifying the coding system and language of on-line documents on the internet.* Proceedings of the 16th International Conference on Computational Linguistics (COLING'1996) (pp. 652-657). Copenhagen, Denmark: ACL, 1996.

KNALLGRAU New Media Solutions. *Java Text Categorizing Library.* Fonte: Source Forge: http://textcat.sourceforge.net/, 2012.

KNOWLEDGE BY DESIGN, Inc. *TextCompactor*. Acesso em: 12 de Fevereiro de 2014, Disponível em: http://www.textcompactor.com/, 2012.

KOEHN, P. *Europarl: A Parallel Corpus for Statistical Machine Translation.* MT Summit 2005, 2005.

KRAHMER, E., MARSI, E., & VAN PELT, P. *Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion.* Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies, 193-196, 2008.

KRUENGKRAI, C., SRICHAIVATTANA, P., SORNLERTLAMVANICH, V., & ISAHARA, H. *Language identification based on string kernels.* Proceedings of the 5th International Symposium on Communications and Information Technologies (ISCIT-2005), pp. 896-899, 2005.

KULKARNI, U. V., & PRASAD, R. S. *Implementation and evaluation of evolutionary connectionist approaches to automated text summarization.* Journal of Computer Science, pp. 1366-1376, 2010.

LEE, D. S., NOHL, C. R., & BAIRD, H. S. *Language identification in complex, unoriented, and degraded document images.* Em: Document Analysis Systems (pp. 17-39). World Scientific, 1998.

LEITE, D., & RINO, L. *Combining multiple features for automatic text summarization through Machine Learning.* Em: A. Teixeira, V. L. Lima, & L. C. Oliveira, Computational Processing of the Portuguese Language: 8th International Conference, PROPOR 2008 (pp. 122-132). Aveiro, Portugal: Springer-Verlag, 2008.

LEXITERIA. *The Lexiteria - Translation, Word Frequency, and N-grams.* Lewisburg, Pennsylvania , USA, 2012.

LIN, C. *ROUGE: a package for automatic evaluation of summaries.* ACL text summarization workshop., pp. 74-81, 2004.

LIN, C.-Y., & HOVY, E. *Automatic evaluation of summaries using n-gram co-occurrence statistics.* Proc. of Human Language Technology Conference (HLT-NAACL 2003), Canada, 2003.

LINS, R. D., & GONÇALVES, P. *Automatic language identification of written texts.* Proceedings of the ACM Symposium on Applied Computing (SAC'04) (pp. 1128-1133). New York, NY, USA: ACM, 2004.

LINS, R. D., SIMSKE, S. J., CABRAL, L. S., SILVA, G. F., LIMA, R. J., MELLO, R. F., & FAVARO, L. *A multi-tool scheme for summarizing textual documents.* Proceedings of 11st IADIS International Conference WWW/INTERNET. Madrid, Spain, 2012.

LITVAK, M., LAST, M., & FRIEDMAN, M. *A New Approach to Improving Multilingual Summarization Using a Genetic Algorithm.* Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics - ACL. Uppsala, Sweden, 2010.

LIU, B., & ZHANG, L. *A Survey of Opinion Mining and Sentiment Analysis.* Em: C. Aggarwal, & C. Zhai, Mining Text Data, pp. 415-463. US: Springer. doi:10.1007/978-1-4614-3223-4_13, 2012.

LIU, X., WEBSTER, J., & KIT, C. *An extractive text summarizer based on significant words.* Proceedings of the 22nd international conference on computer processing of oriental languages. Language technology for the knowledge-based economy, 168-178, 2009.

LLORET, E., & PALOMAR, M. *A gradual combination of features for building automatic summarization systems.* Proceedings of the 12th international conference on text, speech and dialogue, pp. 16-23, 2009.

LLORET, E., & PALOMAR, M. *Text summarization in progress: A literature review.* Artificial Intelligence Review, 37(1), 1-41, 2012.

LLORET, E., & PALOMAR, M. *Compendium: A Text Summarisation Tool for Generating Summaries of Multiple Purposes, Domains, and Genres.* Natural Language Engineering (NLE), 19(2): 147-186, 2013.

LODHI, C. S., SHAWE-TAYLOR, J., CRISTIANINI, N., & WATKINS, C. J. *Text classification using string kernels.* Journal of Machine Learning Research, 2, 419-444, 2002.

LUHN, H. *The automatic creation of literature abstracts.* IBM Journal of, 2(2), 159-165, 1958.

MARIÒO, J., BANCHS, R., CREGO, J., GISPERT, A., LAMBERT, P., FONOLLOSA, J., & AL., E. N-gram-based machine translation. Computational Linguistics, 32(4),, 527-549, 2006.

MARTINS, B., & SILVA, M. J. *Language Identification in Web Pages.* Proceedings of 2005 ACM Symposium on Applied Computing (SAC'05) (pp. 764-768). Santa Fe, New Mexico, USA: ACM, 2005.

MCNAMEE, P., & MAYFIED, J. *Character N-gram Tokenization for European* Language Text Retrieval. Information Retrieval, 7, 73-97, 2004.

MICROSOFT. *Microsoft Translator V2.* (MSDN) Acesso em: 10 de Março de 2014, Disponível em: http://msdn.microsoft.com/en-us/library/ff512423.aspx, 2014.

MIHALCEA, R., & TARAU, P. *TextRank: Bringing order into texts.* Conference on empirical methods in natural language processing, 2004.

MOCANU, D., BARONCHELLI, A., PERRA, N., GONÇALVES, B., ZHANG, Q., & VESPIGNANI, A. *The Twitter of Babel: Mapping World Languages through Microblogging Platforms.* PLoS ONE 8(4): e61981. doi:10.1371/journal.pone.0061981, 2013.

MURDOCK, V. G. *Aspects of sentence retrieval.* Ph.D. thesis. Amherst: University of Massachusetts, 2006.

MUTHUSAMY, Y. K., & SPITZ, A. L. *Automatic language identification.* State of the Art in Human Language, pp. 273-285, 1996.

NENKOVA, A., & MCKEOWN, K. *Automatic summarization.* Foundations and Trends in Information Retrieval, 5(2–3), 103-233, 2011.

NENKOVA, A., & MCKEOWN, K. *A Survey of Text Summarization Techniques.* Em: C. Aggarwal, & C. Zhai, Mining Text Data (pp. 43-76). Springer US, 2012.

NEWMAN, P. *Foreign Language Identification: First Step in the Translation Process.* Proceedings of the 28th Annual Conference of the American Translators Association, pp. 509-516, 1987.

ORASAN, C. *Comparative evaluation of term-weighting methods for automatic summarization.* Journal of Quantitative Linguistics, 16, 67-95, 2009.

PARDO, T., & RINO, L. *TeMario: a corpus for automatic text summarization.* São Paulo: Technical report, NILC-TR-03-09, 2003.

PEDERSEN, T., PATWARDHAN, S., & MICHELIZZI, J. *Wordnet::similarity: Measuring the relatedness of concepts.* Demonstration papers at HLT-NAACL 2004., 38-41, 2004.

POUTSMA, A. *Applying Monte Carlo Techniques to Language Identification.* Proceedings of Computational Linguistics in the Netherlands (pp. 179-189). Amsterdã: CLIN, 2001.

PRASAD, R., UPLAVIKAR, N., WAKHARE, S., JAIN, V., & YEDKE, T. *Feature based text summarization.* International Journal of Advances in Computing and Information Researches, pp. 15-18, 2012.

RADEV, D., ALLISON, T., BLAIR-GOLDENSOHN, S., BLITZER, J., ÇELEBI, A., DIMITROV, S., . . . ZHU, Z. *MEAD - a platform for multidocument multilingual text summarization.* Proceedings of LREC 2004. Lisbon, Portugal, 2004.

ROARK, B., & FISHER, S. *OGI/OHSU baseline multilingual multi-document summarization system.* IEEE International Conference on Microelectronic Systems Education. Anaheim, CA, USA, 2005.

SATOSHI, C., SATOSHI, S., MURATA, M., UCHIMOTO, K., UTIYAMA, M., & ISAHARA, H. *Keihanna human info-communication. Sentence extraction system assembling multiple evidence.* Proceedings 2nd NTCIR workshop, 319-324, 2001.

SCHULZE, B. *Automatic language identification using both n-gram and word information.* (C. Xerox Company Stamford, Ed.) US Patent Number: 6.167.369 (US006168369A), Xerox Company Stamford, Conn, 2000.

SHUYO, N. *Language Detection Library for Java.* Fonte: Google Code: http://code.google.com/p/language-detection/, 2010.

SIBUN, P., & REYNAR, J. C. *Language identification: Examining the issues.* Proceedings of the 5th Symposium on Document Analysis and Information Retrieval, 1996.

SIBUN, P., & SPITZ, L. *Language Determination: Natural Language Processing from Scanned Document Images.* Proceedings of the 4th Applied Natural Language Processing Conference (pp. 15-21). Sttutgard: Germany, 1994.

SPARCK-JONES, K. *Automatic summarising: Factors and directions.* Em: I. Mani, & M. Maybury, Advances in Automatic Text Summarization (pp. 1-12). London: MIT Press, 1999.

TAKAMURA, H., & OKUMURA, M. *Text summarization model based on the budgeted median problem.* 18th ACM Conference on Information and knowledge management problem, pp. 1589-1592, 2009.

TEYTAUD, O., & JALAM, R. *Kernel-based text categorization.* Proceedings. IJCNN '01. International Joint Conference on Neural Networks. 3, pp. 1891 - 1896. Washington, DC, USA: INNS-IEEE. doi:10.1109/IJCNN.2001.938452, 2001.

TIAN, J., & SUONTAUSTA, J. *Scalable neural network based language identification from written texts.* Proceedings of IEEE ICASSP 2003 (pp. I48-I51). Hong Kong, CN: IEEE, 2003.

TONELLI, S., & PIANTA, E. *Matching documents and summaries using key-concepts.* Proceedings of the french text mining evaluation workshop, 2011.

WANG, D., & LI, T. *Document update summarization using incremental hierarchical clustering.* 19th ACM international conference on Information and knowledge management, pp. 279-288, 2010.

WEI, Y. *Document summarization method based on heterogeneous graph.* 9th International conference on fuzzy systems and knowledge discovery (FSKD), 1285-1289, 2012.

ZHANG, D., MA, J., NIU, X., GAO, S., & SONG, L. *Multi-document summarization of product reviews.* 9th International conference on fuzzy systems and knowledge discovery (FSKD), 1309-1314, 2012.

Apêndices

A – Produção acadêmica durante o período de escrita da Tese

Tarefa 1 – Identificação automática de idiomas (Capítulo 2)

1. **CABRAL, L. S.**; LINS, R. D.; LIMA, R. J.; SIMSKE, S. J. *A comparative assessment of language identification approaches in textual documents.* In: IADIS International Conference Applied Computing 2012, 2012, Madrid. Proceedings of IADIS International Conference Applied Computing 2012. Madrid: IADIS, 2012. p. 67-74. (Qualis B2)
2. **CABRAL, L. S.**; LIMA, R. J.; LINS, R. D.; MELLO, R. F.; FREITAS, F.; SILVA, G. F.; SIMSKE, S. J.; FAVARO, L. *A Hybrid Algorithm for Automatic Language Detection on Web and Text Documents.* Booklet of 11th IAPR International Workshop on Document Analysis Systems, 2014. (Qualis B1)

Tarefa 2 – Sumarização automática de textos (Capítulo 3)

1. LINS, R. D.; SIMSKE, S. J.; **CABRAL, L. S.**; SILVA, G. F.; LIMA, R. J.; MELLO, R. F.; FAVARO, L. *A multi-tool scheme for summarizing textual documents.* In: IADIS International Conference WWW/Internet 2012, 2012, Madrid. Proceedings of IADIS International Conference WWW/Internet 2012. Madrid: IADIS, 2012. p. 409-414. (Qualis B2)

Tarefa 3 – Sumarização Extrativa (Capítulo 4)

1. MELLO, R. F.; **CABRAL, L. S.**; LINS, R. D.; SILVA, G. F.; FREITAS, F.; LIMA, R. J.; SIMSKE, S. J.; FAVARO, L. *Assessing Sentence Scoring Techniques for Extractive Text Summarization.* Expert Systems with Applications, v. 41, p. 3082-3094, 2013. (Qualis A1)

Tarefa 4 – Sumarização Independente de Idioma (Capítulo 5)

1. **CABRAL, L. S.**; LINS, R. D.; MELLO, R.; FREITAS, F.; ÁVILA, B.; SIMSKE, S.; RISS, M. *A Platform for Language Independent Summarization.* In: Proceedings of the 14th ACM Symposium on Document Engineering 2014 (DocEng'14) Fort Collins, Colorado, US. September 16-19, 2014. (Qualis B1)

A COMPARATIVE ASSESSMENT OF LANGUAGE IDENTIFICATION APPROACHES IN TEXTUAL DOCUMENTS

Luciano de Souza Cabral[1,2], Rafael Dueire Lins[1], Rinaldo Lima[1], Steven J. Simske[3]

[1] *Federal University of Pernambuco, Recife, Brazil*
[2] *Federal Institute of Pernambuco, Caruaru, Brazil*
[3] *Hewlett-Packard Labs., Fort Collins, CO 80528, USA*

ABSTRACT

This paper presents several experiments conducted for assessing distinct methods for language identification of written texts. After introducing a new method for the language identification problem, we conducted some standard experiments aiming at evaluating the proposed approaches against three other ones. In order to perform fair comparisons, we used the same corpus (EuroParl Corpus), which contains 21,000 sentences evenly distributed in 21 languages. We discuss the experimental results as well as the strengths and limitations of the compared algorithms. In addition, the accuracy results achieved by the proposed method introduced in this research work showed that it is very competitive with other state of the art methods.

KEYWORDS

Language Identification, Comparative Analysis, Document Engineering.

1. INTRODUCTION

Automatic language identification is used to determine the primary idiom of written or audio content. Some survey papers, such as (Sibun & Reynar, 1996) and (Hughes et al., 2006), address this problem in the literature, reaching different conclusions about the efficiency of the presented methods. Although the initial steps in this research area date back to the mid 1960s, there are still open questions and the appearance of the Internet has brought new challenges to the field.

Language identification in textual documents has been addressed by many researchers using distinct approaches and techniques. The applicability of such techniques has gained greater importance in the context of search engines for the Internet, as they are fundamental in accurate information retrieval. Despite all the scientific and industrial efforts, a detailed analysis and assessment of such strategies on uniform corpora remains unpublished, and may reveal new research challenges.

This work attempts to answer some questions focusing on the problem of identifying the language used in written single-idiom documents. More particularly, we want to evaluate some language identification methods. In this evaluation, the two most important criteria used for assessing them will be both accuracy (correct classification rate) and processing time. In order to have a fair comparison among the selected methods, we conduct the evaluation experiments using the same competition corpus, and the same experimental setup.

This paper is structured as follows: Section 2 presents the related work. Section 3 describes the four selected methods that deals with the problem of automatic language identification. The grounds for a fair benchmarking and experimental results are presented and discussed in Section 4. The paper closes by presenting the conclusions and outlining future work in Section 5.

2. RELATED WORK

Pioneering work in automatic language identification was published by E.M.Gold in 1967 (Gold, 1967) who proposed the analysis of closed grammatical classes using a list of languages. Sibun and Reynar (1996) present a valuable and unbiased survey of the techniques described before 1996. The state of the art then was based on statistical co-occurrence based on n-grams and Linear Discriminant Analysis. In the field of computational linguistics, an n-gram is a contiguous sequence of n items (characters or words) from a given sequence of text.

More recent work reported in the literature (Hughes et al., 2006) also brings a valuable survey of the advances in the field before 2006 emphasizing the use of derivations of probabilistic Bayesian and Markov models by Dunning (Dunning, 1994); the application of models based on word vectors (Danarshek, 1995); and the language identification and context retrieval using n-grams as suggested by McNamee and Mayfield (McNamee & Mayfield, 2004).

Reference (Bhargava & Kondrak, 2010) presents a method using Support Vector Machines (SVM) for language identification of very short texts such as proper nouns. They showed that SVMs outperform language models on two different data sets consisting of personal names.

Teytaud and Jalam (Teytaud & Jalam, 2001) apply the kernel method with n-grams obtained by Inverse Document Frequency. Lodhi and his colleagues (Lodhi et al., 2002) proposed a method that prefers strings instead of words for generating the kernel, with promising results for texts of different languages. Poutsma (2001) introduced a technique for language identification based on Monte Carlo sampling. He demonstrated that, by determining the language of a large enough number of random features, one can determine the document language to be the language which result most often from these features. Whether the amount of samples is sufficiently large can be determined by calculating the standard error of the samples.

The topic of language identification may be considered fashionable today, as three papers have already been published in 2012. The first by Botha and Barnard (Botha & Barnard, 2012), analyzed the factors that affect the precision of language identification in textual documents. Kamper and Niesler (Kamper & Niesler, 2012) presented a survey in dialect, language, and accent identification. Geiger and colleagues (Geiger et al., 2012) present an approach with reduced-size n-grams.

Lins and Gonçalves (Lins & Gonçalves, 2004) emphasized computational performance by selecting the highest accuracy of a tree of cascaded classifiers applied on *closed grammatical classes*[1]. Their classifiers were tested on a statistical relevant corpus composed of plain texts and web pages.

Two other papers (Martins & Silva, 2005) (Kikui, 1999) also treated web pages as a corpus for the language identification problem. More particularly, Martins and Silva (Martins & Silva, 2005) described a system to automatically identify the language of web pages that is equally based on the n-gram model originally proposed in (Cavnar & Trenkle, 1994). The author's contribution consists in a more efficient similarity measure, as well as some additional heuristics to handle web data.

3. DESCRIPTION OF THE COMPARED SYSTEMS.

This section describes the four methods for language identification participating in the comparison provided in this work, namely (iii) *LG*, (ii) *CALIM*, (iii) *TextCat*, and (iv) *Language Detector*. Particularly, we examine in detail our method for language identification (CALIM), presenting its underlying assumptions as well as giving an explanation of its algorithm.

LG. The algorithm presented in Lins and Gonçalves (Lins & Gonçalves, 2004) makes use of some closed grammatical classes in several languages as a fast way to identify the language of a document. A tree of cascade classifiers is employed. The first one counts the number of English adverbs, the second counts the number of Spanish prepositions, etc. Such closed-class prioritization was decided a priori by statistical analysis of several documents. The original paper by Lins and Gonçalves (Lins & Gonçalves, 2004) covers only 6 languages, thus it had to be extended following the method originally described in (Lins & Gonçalves,

[1] *closed grammatical classes* are grammatical categories that have no flections, such as prepositions, adverbs, conjunctions, etc.

2004) to cover all the 21 languages in the Europarl corpus. This system, referenced here as *LG*, forms a dictionary prioritizing some closed grammatical subclasses for each language, based on a statistical study of the relative frequency of occurrence of words in texts for each language. The language of the text is determined after scanning the input document looking for words in the dictionaries. If at least 5 words are found in one of the dictionaries, provided the relative frequency is at least 40% superior than the second dictionary, one says that the document is written in the language associated with the top-ranked dictionary. Such decision criteria was removed in this version as the test corpus is a competition one and some document entries are as small as 1-phrase long. Instead, the language that presents the highest number of entries is the detected language of the document, straight away. Singular lexical patterns such as "ão" are particular of Portuguese (as in "não"). The inclusion of such pattern detection increases the capacity of language identification of the algorithm.

CALIM. This new method for language identification was inspired on the ideas of Dunning (Dunning, 1994) and Lins and Gonçalves (Cavnar & Trenkle, 1994). CALIM is based on language profile dictionaries that take into account frequent short words present in all languages under study (21 in total). More precisely, the creation of such dictionaries takes into account approximately the 250 more frequent words for each language. For creating these language profiles, we used some dictionary databases provided by Lexiteria which is an initiative aiming at understanding various aspects of human language (Lexiteria, 2002). Lexiteria provides specialized word lists, including word frequency lists (some of them with part of speech) as well as glossaries, and custom dictionary databases. For our research purposes, we collected some statistics for 21 languages, such as word frequency, average word length, etc. After creating the language dictionaries, sorted by word frequency in decreasing order, we select the first *M* words, given by the following formula.

$$M = \left(\frac{\sum_{i=1}^{n} length(w_i)}{n} \right)^2 \times 10. \qquad (1)$$

The underlying assumption CALIM takes at this point is that it will be more probable to find high frequency words in the input text than low frequency words. In addition, due to performance reason, we only consider words at maximum length 5 ($n = 5$). We justify this choice because, in most languages, the most frequent words like prepositions, personal pronouns, etc. are also the shorter ones. Thus, in case a word is longer than 5 characters, we take its 5-length suffix to be included in the language dictionary. We considered other values for n in our experiments, but we achieved the best performance results with $n = 5$.
During the classification step of CALIM, we applied a very useful heuristic that contributed to more accurate results in our experimental evaluation. This heuristic assumes that if a word (or token) comprises very specific n-grams exclusively found in certain language (like "ão" in Portuguese language), then the method assigns a greater value than it is done in normal voting schema which is equal to 1.
We adopted the simple criteria that the language with the highest accumulative scoring value will be chosen. At the end of the previous classification step, each token from the input text will be classified in 1 or more languages. In case of two of more languages final score ended in a draw, we proceed with an additional scoring step consisting in multiplying the final score of each token by the normalized frequency of that token. The normalized frequency of a token is simply calculated by the ratio between the token frequency and the sum of all token frequencies for a language dictionary. This simple heuristics contributed to choose the right language of the document.

TextCat. There are several tools and platforms for language identification in texts, some well known and referenced by most of the papers listed in the last section. Amongst the best-known and used tools, one finds the Java Text Categorizing (TextCat)[2] library (Knallgrau, 2005). It consists of a pure Java implementation of

[2] http://textcat.sourceforge.net/

the *LibTextCat*[3] library for language identification written in C. TextCat is distributed under the LGPL[4] and can also be used for categorizing text into arbitrary topics by computing appropriate fingerprints which represent the categories, as it was originally proposed in (Cavnar & Trenkle, 1994). This algorithm adopts the n-gram model for representing a document, which seems to be the most promising approach. The central idea of this algorithm is to calculate a fingerprint of a document associated to an unknown category, and compare it with the fingerprints of a number of documents of which the categories are known. A fingerprint is a list of the most frequent n-grams occurring in a document, ordered by frequency. Fingerprints are compared by using a simple out-of-place computation (Cavnar & Trenkle, 1994). Finally, the categories of the closest matches are output as the classification.
The primary advantage of this algorithm is the claimed support for language identification of noisy texts, e.g., texts coming from OCR systems. Among its disadvantages, it is noteworthy that the formation of profiles requires time and it lacks support for important languages such as Portuguese.

Language Detector. It consists of a library for language identification developed in Java under Apache open license. It was implemented in by Shuyo (Shuyo, 2010) based on the techniques proposed by Dunning (Dunning, 1994). In his approach, Dunning assumes that language can be modeled by a low order Markov process which generating tokens, and then using Bayesian decision rules for classifying them. Moreover, the author also claims have 99% average accuracy in discriminating two moderately related languages, English and Spanish.

4. EXPERIMENTAL SETUP

The performance evaluation of four methods for automatic language identification is assessed here. After presenting the experimental setup adopted in benchmarking the methods, we discuss the achieved results.

4.1 Testing Environment

All research in language identification in written texts focuses on a restricted number of languages to be identified, which restricts the size of the domain to be studied because of the corpora used. As very few works make their corpora available for other researchers, it becomes difficult the assertion for sure the validity of the claimed results.

In this context, this work assesses four distinct language identification algorithms in a fair and uniform manner. All algorithms were implemented in the same programming language and tested on the same corpus. Amongst the several available corpora, the European Parliament Proceedings Parallel Corpus or EuroParl Corpus (Koehn, 2005) deserved special attention for our research purposes. This corpus is composed of documents reporting speeches and discussions occurred in the European Parliament since 1996. This corpus encompasses document in 21 languages of the European Community and it is was used in competitions sponsored by international associations and conferences, such as the Association for Computational Linguistics (ACL), the Conference on Empirical Methods on Natural Language Processing (EMNLP), and the Workshop on Machine Translation (WMT), just to mention a few.

The benchmarking reported here adopted the plain text document format of the EPPPC corpus, which comprises 21,000 documents collected from the 1996-2011 period, equally divided among 21 European languages.

In order to yield a fair time performance analysis, all the above approaches either were implemented in Java or used an available implementation in the same language.

In addition, we measured the elapsed time (in seconds) it took to run each experiment. These measures were taken using a Core 2 Duo processor (1.83 GHz) machine equipped with 4GB RAM, running the Windows operating system (64 bits version).

---

[3] http://software.wise-guys.nl/libtextcat/
[4] GNU Lesser General Public License, http://www.gnu.org/copyleft/lesser.html

4.2 Results and Discussion

For all experiments reported in this section, we took into account the fairness of the comparison ground. Table 1 provides the average accuracy results obtained for all algorithms evaluated on two subsets of the EPPC Corpus composed of five[5] and thirteen[6] languages.

First we have considered to extend the LG system to identify more 15 languages, i.e., 21 languages found in the Europarl corpus minus the 6 languages originally covered by LG (Lins & Gonçalves, 2004). Since the LG dictionary is based on closed grammatical classes of words, we had difficulty in finding such kind of dictionaries for these complementary languages. As a possible solution to that, we used the Wiktionary[7], a collaborative project to produce a free-content multilingual dictionary, as a source for the grammatical classes of words we were interested in.

According to the Table 1, although the LG language identification method performed relatively well with 5 languages, its accuracy drops dramatically with 13 languages. One possible reason for explain such behavior, as already mentioned in Section 3, is that this method was originally "tuned" for only 6 languages, whereas the CALIM and Language Detector can handle more languages. One positive aspect of the LG method is its lower CPU time usage (in second) among all the selected methods. This is explained by the fact that the LG method has a much smaller dictionary compared with the other methods, which means less dictionary entries to deal with. These results suggest that with an appropriate learning procedure for extend the language dictionaries (or profiles), the LG method may be a good choice if response time is the most important concern.

The other methods have not present statistically significant difference against each other with respect to accuracy scores for 5 languages, yielding indeed high performance, but the processing time varied greatly. In our experiments, the TextCat implementation of the Cavnar and Trenkle's method for language identification took much more time (Tables 1 and 2).

Table 1. Average results for 5 and 13 languages (Europarl Corpus)

Methods	5 languages		13 languages	
	Accuracy	CPU time (s)	Accuracy	CPU time (s)
Lins and Gonçalves (LG)	93.68	3.78	64.74	9.83
TextCat (TC)	97.10	137.87	88.37	358.47
CALIM (CL)	99.02	3.01	96.36	7.08
Language Detector (LD)	99.48	3.82	98.92	9.95

Table 2. Average results for 6 (Europarl corpus)

Methods	6 languages		21 languages	
	Accuracy	CPU time (s)	Accuracy	CPU time (s)
Lins and Gonçalvez (LG)	94.72	4.54	-	-
CALIM (CL)	98.93	3.60	97.08	12.62
Language Detector (LD)	99.45	4.59	99.16	16.08

The experimental results shown in Tab. 2, we did not consider the TextCat because of its limited number of supported language. Thus, the experimental results with the other remaining methods for 6 languages, including the Portuguese language, are quite similar to those reported in Table 1. Analogously, we do not report the results for the LG method in the 21 language test setting, because its language dictionaries do not cover languages such as Estonian, Polish and Slovak in the test corpus.

The remaining CL and Language Detector methods obtained high accuracy scores for the test samples comprising the whole set of 21 languages. The LD method achieved more than 99% of accuracy requiring only 16 seconds to process 21,000 text samples. Even faster was the CL method, but with a slightly lower accuracy.

[5] German, English, Spanish, French, Italian
[6] Danish, German, English, Spanish, Finnish, French, Hungarian, Italian, Dutch, Polish, Slovakian, Slovenian, Swedish.
[7] http://en.wiktionary.org/wiki

The hybrid method CL proposed in this paper presented a competitive performance with respect to accuracy performance, but it took less CPU time than LD in all experiments (Tab. 1-4).

One factor that can hamper the identification accuracy is related to the corpus size, which still presents a challenge for language identification methods when the input text is short. Thus, aiming at assessing the sensitivity of the selected methods in function of the length (in characters) of the input text sample, we have performed two other experiments. Tables 3 and 4 present both the accuracy and CPU time obtained for 13 and 21 languages text samples, respectively.

Another aspect we wanted to evaluate has to do with the influence of the suffixes as elements in the language dictionaries. For that, we conducted another experiment with 21 languages considering the dictionary creation without the suffixes. The results revealed that the algorithm loses 5.1% in accuracy, but its processing time is also reduced in 3.3 seconds. This was expected since the algorithm processes less tokens.

The analysis concerning the Tables 3 and 4 reveals us that TC did not have considered performance aspects in its java implementation, being the slowest language identification prototype in our evaluation. Only taking 50 input characters, LD achieved accuracy of 98% for 13 languages, and 97% for 21. On the other hand, CALIM, for the same languages, obtained 90% and 92% as accuracy rates. This serves as evidence that both methods require short text input for achieving very good results.

Figure 3 shows the confusion matrix for the dataset with 21 languages in which one can distinguish the false positive results obtained by CALIM. The only anomaly we have detected was concerning the high number of false positives between the Romanian and Slovakian language in 135 test examples, indicating the Romanian language as the correct one in this case. The main reason for that is because these two languages have many words and suffixes in common, e.g., "to, sa, je".

Indeed, during the computation of the token scores, the CALIM algorithm takes into account the language with the higher relative frequency among the language dictionaries, prioritizing the Romanian language in this case. In future work, we plan to analyze the effect of removing the word intersection among the dictionaries with the aim of avoiding such type of misclassifications. Another possibility is the definition of another strategy to tie-break in case of an even final score for two or more languages. Other false positive scores (27 for Polish and Hungarian) and (22 for Slovakian and Czech) were originated by the same reason, but with fewer cases.

Finally, the last column(nn = none) in the confusion matrix indicates the number of documents for a given language that the CALIM algorithm was not able to classify. This happened due to the absence of any token or token suffix of the input text in language dictionaries. In fact, we detected very short text encompassing only 4 tokens, like "Mødet åbnet kl. 09.00" for Danish.

Table 3. Sensibility analysis results - (CPU time usage and accuracy rate for 13 languages)

Length	Accuracy			CPU time		
(char)	TC	LD	CL	TC	LD	CL
10	0.54	0.65	0.47	42.78	11.97	4.62
20	0.69	0.85	0.69	73.44	11.59	4.81
30	0.76	0.93	0.80	107.96	11.60	5.18
40	0.80	0.96	0.86	132.83	11.50	5.26
50	0.82	0.98	0.90	175.75	11.49	5.67
75	0.84	0.99	0.94	242.10	11.99	6.55
100	0.85	0.99	0.96	272.92	11.22	6.59

Table 4. Sensibility analysis results - (CPU time usage and acurracy rate for 21 languages)

Length (char)	Accuracy		CPU time	
	LD	CL	LD	CL
10	0.68	0.53	15.08	6.92
20	0.86	0.74	15.94	7.72
30	0.93	0.84	15.40	7.88
40	0.96	0.89	15.06	8.39
50	0.97	0.92	15.52	8.78
75	0.98	0.96	15.45	9.58
100	0.99	0.97	15.41	10.36

CALIM - Confusion Matrix – 21 languages

	bul	cze	dan	ger	gre	eng	spa	est	fin	fre	hun	ita	lit	lat	dut	pol	por	rom	slvk	slvn	swe	nn
bul	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cze	0	967	0	1	0	1	1	0	0	0	2	0	0	0	0	4	1	4	17	0	0	2
dan	0	0	995	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0'	1	1
ger	0	0	1	992	0	0	0	0	0	1	0	0	0	0	0	4	1	1	0	0	0	0
gre	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
eng	0	0	2	0	0	992	3	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0
spa	0	0	0	0	0	0	996	0	0	0	0	0	0	0	0	2	1	0	0	0	0	1
est	0	0	0	1	0	1	1	944	6	11	8	1	0	0	0	13	0	1	1	0	0	12
fin	0	0	0	0	0	0	1	13	937	5	7	1	2	1	0	20	0	0	0	0	0	13
fre	0	0	0	0	0	0	6	0	0	986	0	0	0	0	1	3	2	0	0	0	0	2
hun	0	0	0	1	0	0	3	0	0	0	952	3	0	0	0	27	7	1	0	0	0	6
ita	0	0	0	0	0	0	9	0	0	1	0	985	1	0	0	2	2	0	0	0	0	0
lit	0	0	0	0	0	0	2	1	0	0	0	1	975	2	0	13	2	0	0	0	0	4
lat	0	0	0	0	0	1	0	0	0	0	0	0	0	995	0	0	0	0	0	0	0	4
dut	0	0	0	1	0	0	0	0	0	0	2	1	0	0	996	0	0	0	0	0	0	0
pol	0	0	0	1	0	0	0	0	0	1	0	0	3	0	5	986	0	1	0	0	0	3
por	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	2	985	0	0	0	0	0
rom	0	0	0	0	0	0	4	0	0	2	0	0	0	0	0	0	0	993	0	0	0	1
slvk	0	22	0	4	0	0	1	0	0	2	0	0	2	0	0	9	0	135	815	4	0	6
slvn	0	20	3	0	0	3	2	1	0	1	2	2	2	0	0	16	6	17	0	917	0	8
swe	0	0	11	1	0	2	1	1	0	0	0	0	0	0	1	1	1	0	0	0	978	3

Figure 3. CALIM Confusion Matrix for the classification of 21 languages

Legend: bul =Bulgarian, cze = Czech, dan = Danish, ger = German, gre = Greek, eng = English, spa = Spanish, est = Estonian, fin = Finnish, fre = French, hun = Hungarian, ita = Italian,lit = Lithuanian, lat = Latvian, dut = Dutch, pol = Polish, por = Portuguese, rom = Romanian, slvk = Slovakian, slvn = Slovenian, swe = Swedish // nn = none

5. CONCLUSION AND FUTURE WORK

This paper presented an assessment of some n-gram based approaches to the automatic language identification problem of written texts. The importance of this reported analysis rests on the fact that all the referenced algorithms were implemented in the same hardware/software platform and tested on the same corpus, allowing, to some extent, a fair comparison between them. Two of the algorithms analysed, namely Language Detector and CALIM, have shown competitive performance not only due to the high precision, but also because of the fast processing time.

The assessment work reported here only scratches the surface. A deeper analysis of typical lexemes, such as the "ão", "ões" in Portuguese may be a simple and efficient way of making further improvements to the

CALIM algorithm, as it does not include similar particularities of other languages. The confusion matrices obtained in our experiments, but not reported here, will be carefully analyzed in order to provide clues for the reasons of misclassifications. For instance, we plan to analyze the effect of removing the word intersection among the dictionaries with the aim of avoid such type of misclassifications.

ACKNOWLEDGEMENT

This research was partly sponsored by the National Council for Scientific and Technological Development (CNPq/Brazil).

REFERENCES

Bhargava, A. and Kondrak, G. 2010. Language identification of names with SVMs. *Human Language Technologies. The 2010 Annual Conference of the North American Chapter of the ACL* (Association for Computational Linguistics), pp. 693–696, Los Angeles, California, June.

Botha, G. and Barnard, E. 2012. *Factors that affect the accuracy of text-based language identification*. Computer Speech & Language, 16 January.

Cavnar, W. B. and Trenkle, J. M. 1994. N-Gram Based Text Categorization. *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*: pg. 161-169.

Danarshek, M. 1995. *Gauging similarity with n-grams: Language independent categorization of text*. Science, pp. 843-848.

Dunning, T. 1994. *Statistical identification of language*. Technical Report CRL MCCS-94-273, New Mexico: Computer Research Lab, New Mexico University.

Geiger, W.; Rauch, J.; Mair, P. e Hornik, K. 2012. *Text Categorization in R: A Reduced N-gram Approach*. Challenges at the Interface of Data Analysis, Computer Science and Optimization. Studies in Classification, Data Analysis and Knowledge Organization. Springer.

Gold, E. M. 1967. *Language identification in the limit*. Information and Control. 1967. pp.447-474.

Hughes, B.; Baldwin, T.; Bird, S.; Nicholson, J. e Mackinlay, A. 2006. Reconsidering language identification for written language resources. *Proceedings of LREC2006*. pp.485-488.

Kamper, H. and Niesler, T. 2012. *A literature review of language, dialect and accent identification*. Technical Report SU-EE-1201. Digital Signal Processing Laboratory. Department of Electrical and Electronic Engineering. Stellenbosch University, South Africa.

Kikui, G-I. 1999. Identifying the coding system and language of on-line documents on the Internet. *Proceedings of the 16th International Conference on Computational Linguistics (COLING 1999)*.

Knallgrau. 2005. *Java Text Categorizing Library*. http://textcat.sourceforge.net/.

Koehn, P. 2005. *Europarl: A Parallel Corpus for Statistical Machine Translation*. MT Summit.

Lexiteria. 2002. *Word Frequency Lists*. http://www.lexiteria.com/.

Lins, R.D. and Gonçalves, P. 2004. Automatic language identification of written texts. *Proceedings of the ACM Symposium on Applied Computing (SAC'04)*.

Lodhi, C. S.; Shawe-taylor, J;. Cristianini, N. e Watkins, C.J.C.H. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419-444.

Martins, B. and Silva, M. J. 2005. Language Identification in Web Pages. *Proceedings of 2005 ACM Symposium on Applied Computing (SAC'05)*, Santa Fe, New Mexico, USA, pp. 764-768.

McNamee, P. and Mayfield, J. 2004. *Character N-gram Tokenization for European Language Text Retrieval*. Information Retrieval, 7:73–97.

Poutsma, A. 2001. Applying Monte Carlo Techniques to Language Identification. *Proceedings of Computational Linguistics in the Netherlands (CLIN)*.

Shuyo, N. 2010. *Language Detection Library for Java*, http://code.google.com/p/language-detection/.

Sibun, P. and Reynar, J.C. 1996. Language identification: Examining the issues. *Proceedings of the 5th Symposium on Document Analysis and Information Retrieval*.

Teytaud, O. e Jalam, R. 2001. Kernel-based text categorization. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

A Hybrid Algorithm for Automatic Language Detection on Web and Text Documents

Luciano Cabral [a], Rinaldo Lima [a], Rafael Lins [a], Fred Freitas [a], Rafael Ferreira [a], Gabriel Silva [a], George Cavalcanti [a], Steven Simske [b] and Marcelo Riss [c]

[a] *Informatics Center, Federal University of Pernambuco, Recife, Brazil*

[b] *Hewlett–Packard Labs., Fort Collins, CO 80528, USA*

[c] *Hewlett–Packard Brazil, Barueri, Brazil*

{lsc4,rjl4,rdl,fred,rflm,gfps,gdcc}@cin.ufpe.br, {steven.simske, marcelo.riss}@hp.com

***Abstract* — Automatic Language Detection is a research area that has gained importance with the Internet and plays a key role in information retrieval. This paper presents a hybrid algorithm to automatic language detection. Our approach relies on classical techniques such as n-gram text analysis, relative frequency and dictionaries of closed-class words. The proposed method is very fast and accurate if compare with its competitors.**

Keywords—language detection; language identification; document engineering; comparative analysis; assessing tecniques

I. INTRODUCTION

The task of automatic language detection has recently emerged as of crucial importance because web search engines have to collect and show multilingual content to the user. Language independent search, well performed by Google, is another good example of application of specific algorithms for automatic language identification. Automatic language identification can be used as a first step towards automatic translation, which certainly increases its usability on the Web.

Some survey papers, such as [1] and [2], address this problem reaching different conclusions about the efficiency of the methods currently available. Although the initial steps in this research area date back to the mid 1960s, there are still open questions and the "birth" of the Internet has brought new challenges to the field.

This work attempts to answer some questions focusing on the language detection problem. More particularly, we want to evaluate a hybrid method called CALIM, which combines three different techniques (Dunning [3], Cavnar and Trenkle [4] and Lins and Gonçalves [5]) that gained new support to web documents. In addition, we implemented other three classical methods in the same language (Java) in order to have a fair and independent performance evaluation.

II. THE CALIM ALGORITHM

CALIM is based on language profile dictionaries that take into account frequent short words present in all languages under study (21 in total). More precisely, the creation of such dictionaries takes into account approximately the 250 more frequent words for each language. For creating such language profiles, we used some dictionary databases provided by Lexiteria [6], which is an initiative aiming at understanding various aspects of human language.

For our research purposes, we collected some statistics for 21 languages, such as word frequency, average word length, etc. After creating the language dictionaries, sorted by word frequency in decreasing order, we selected the top frequent *250* words.

The underlying assumption in CALIM is that the selected high frequency words are more likely to be found in the input text than low frequency ones. In addition, due to performance reasons, we only consider words that have maximum length *5* characters ($n = 5$). We justify this choice because, in most languages, the most frequent words like prepositions, personal pronouns, etc. are also the shorter ones.

During the classification step of CALIM, we applied a heuristic that contributed to more accurate results in our performance evaluation. That heuristic assumes that if a word (or token) comprises very specific n-grams exclusively found in certain language (like "*do*" in the Portuguese language), then the method assigns a greater value to it than it is done in normal voting schema in which the normal vote is equal to *1*.

The normalized frequency of a token is simply calculated by the ratio between the token frequency and the sum of all token frequencies for a language dictionary. This simple heuristics seemed to contribute to determine the correct language of the document.

A. Other Implementations

In addition to the CALIM algorithm, we decided to develop the *simple closed-class dictionary* method proposed by Lins and Gonçalves [5] (such method is here labelled *CALIG)*, and the Language Detector [7], [3], labelled here as *LangDetect*.

- *SimpleDic*: It is a simple dictionary method, developed using as the main component, the common stop words of, commonly ignored in some Natural Language Processing (NLP) procedures. Those stop words were useful in the dictionaries formation as they are short, simple, and most of times invariant to gender or quantity (singular or plural).
- *LangDetect:* Language Detector consists of a library for language identification developed in Java. It was implemented in [7] by Shuyo, based on the techniques proposed by Dunning [3].
- *CALIG*: We revisited the work of Lins and Gonçalves [5] and used their methodology to develop new dictionaries of closed classes from the 6 original ones to 25 languages in total, using two layers of treatment. In the first layer, the lexical analyser recognizes the most common lexical features, i.e. "*ñ*" for Spanish, "*ß*" for German, "*æ*" for Danish, etc. The second layer takes into account the closed class dictionaries and a decision

The research results reported in this paper have been partly funded by a R&D project between Hewlett-Packard Brazil and UFPE originated from tax exemption (IPI-Law n 8.248, of 1991 and later updates).

tree heuristic to choose the best language using the ratio of the total number of tokens in the document divided by the number of recognized tokens in the document.

As already mentioned, the original paper by Lins and Gonçalves [5] covers only 6 languages, thus we had to extend it following the method originally described in [5] to cover all the 21 European languages plus four other languages with lexical features, e.g. Arabic, Hebrew, Hindi and Korean.

III. Experimental Setup

The test bed used here is an extension of the one described in reference [2]. The new test environment was adapted to recognize the format of files from the web, i.e., HTML and XML, removing tags and annotations from them, leaving for analysis only what really matters for the scope: the text part.

A. Environment

In our experiments, we used a laptop equipped with a processor Intel Core i3-2330M 2.20 Ghz, 4 GB RAM, and the Microsoft Windows 7 operating system. We chose Java as development language and accessing serialized files.

B. Test Corpora

We used the Europarl v7 corpus [8] that has two versions, the first one called "*test*" which is composed of 21,000 documents containing very small size texts. This dataset presents an equal distribution between documents and languages supported (1,000 documents per language). The second one, called "*full*", with about 60,000 some size XML documents, with a random distribution between documents and languages (Bulgarian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, and Swedish).

C. Results and discussion

We processed the *Europarl* Test Corpus, and observed similar results compare to the ones in [2]. We have performed two test set ups, one with the 21 European languages (already mentioned), and another with the 6 most common web languages (English, French, German, Italian, Portuguese and Spanish). Table I summarizes the obtained results.

TABLE I. Average results for 6 and 21 languages (Test)

Methods	6 languages		21 languages	
	Acc	Time(s)	Acc	Time(s)
SimpleDic	94.72	3.65	63.23	12.79
CALIM	98.93	2.87	97.08	10.06
LangDetect	99.48	3.25	99.16	11.38
CALIG	92.13	12.78	92.42	44.79

According to Table I, the best performance was obtained on the *Test Corpus* in which *LangDetect* (more accurate, with 0.55 and 2.08 percent higher than *CALIM* on 6 and 21 languages respectively), and *CALIM* (faster, with 0.38 and 1.32s smaller than *LangDetect* to 6 and 21 languages respectively).

TABLE II. Average results for 6 and 21 languages (Full)

Methods	6 languages		21 languages	
	Acc	Time(s)	Acc	Time(s)
SimpleDic	100.00	871,76	80.123	3051.16
CALIM	100.00	793.37	99.992	2776.79
LangDetect	100.00	910.32	99.993	3186.12
CALIG	99.97	676.97	99.942	2369.40

Analyzing Table II, the result with the Full Corpus, that is more appropriate to the reality of any web document, the CALIG strategy was faster, with shorter processing time then CALIM and LangDetect (which had similar accuracy) in 14.67% and 25.63%, respectively, above the best time.

In addition, the accuracy for 6 and 21 languages in just 0.03 and 0.051 percentage points below the bests (*CALIM* and *LangDetect*). Thus, the results suggest that, in terms of accuracy and processing time, CALIM and CALIG strategies obtained the best performance on the *Europarl Test* and *Full Corpora* datasets, with less processing time, and higher accuracy.

IV. Conclusion and Future Work

This paper presented an assessment of some automatic language detection techniques, and most importantly, a hybrid algorithm inspired on the ideas of Dunning [3], Cavnar and Trenkle [4] and Lins and Gonçalves [5]. The importance of the reported analysis rests on the fact that all the referenced algorithms were implemented in the same hardware and software platform, and assessed on the Europarl corpus at different versions allowing a fair comparison amongst them.

Two of the algorithms analyzed on the Test Corpus (Plain text) experiment, namely LangDetect and CALIM, have shown competitive performance not only due to the higher accuracy, but also faster processing time. The former was more accurate, the latter was faster. On the Full corpus experiment, the aforementioned algorithms appear with the better results in terms of accuracy, with virtually the same performance score on this second dataset, and CALIM still running faster than LangDetect.

The CALIG algorithm strategy appears as faster in the Full corpus experiment. This result is interesting, because it has a larger quantity of dictionaries implemented (133 hash table dictionaries). The other strategies not exceed 53 dictionaries. For future work, we plan to: (i) solve common problems, such as multilingual document classification; (ii) integrate some summarization strategies aiming at language independent summarization tasks.

References

[1] B. Hughes, T. Baldwin, S. Bird, J. Nicholson and A. Mackinlay, "Reconsidering language identification for written language resources," *Proceedings of LREC 2006*, pp. 485-488, 2006.

[2] L. Cabral, R. Lins, R. Lima and S. Simske, "A Comparative Assessment of Language Identification Approaches in Textual Documents.," *Proceedings of IADIS Applied Computing 2012*, July 2012.

[3] T. Dunning, "Statistical identification of language," Technical Report CRL MCCS-94-273, Computer Research Lab, New Mexico University, New Mexico, 1994.

[4] W. B. Cavnar and J. M. Trenkle, "N-Gram Based Text Categorization.," *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161-169, 1994.

[5] R. Lins and P. Gonçalves, "Automatic language identification of written texts.," *Proceedings of the ACM Symposium on Applied Computing (SAC'04)*., 2004.

[6] Lexiteria, " Word Frequency Lists.," Lexiteria, 2002. [Online]. Available: http://www.lexiteria.com/. [Accessed 09 10 2013].

[7] N. Shuyo, "Language Detection Library for Java," 2010. [Online]. Available: http://code.google.com/p/language-detection/. [Accessed 8 October 2013].

[8] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," *MT Summit 2005*, 2005.

A MULTI-TOOL SCHEME FOR SUMMARIZING TEXTUAL DOCUMENTS

Rafael Dueire Lins[1], Steven J. Simske[2], Luciano de Souza Cabral[1], Gabriel de França Silva[1], Rinaldo Lima[1], Rafael F. Mello[1], Luciano Favaro[3]

[1]Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil
[2]Hewlett-Packard Labs., Fort Collins, CO 80528, USA
[3]Hewlett-Packard do Brasil, Barueri, SP, Brazil

ABSTRACT

Sentence-based extractive summarization is a promising and widely implemented option for efficient text summarization. Different research groups all over the world have made their summarization tools available in the Internet either for downloading or for on-line use. Because of the plurality of summarizers, a logical approach to improving overall summarization is to find a means to effectively combine the output of multiple summarizers. This paper presents a new approach for single text summarization which uses the results of different tools to generate a single summary.

KEYWORDS
Text summarization, single documents, extractive summarization, document engineering.

1. INTRODUCTION

Increasing online availability of text-based documents on Sharepoints, private and public clouds, websites, social websites, etc., has prompted the need for efficient and accurate data mining approaches suitable for textual data analysis. Being able to automatically discover which area of knowledge a given document belongs to is a challenging task. Being able to automatically summarize its content is a complex information mining job. Spark-Jones [11] defined summarization as a reductive transformation of source text to summary text through content condensation by selection and/or generalization of what is considered important in the source. Research on summarization started in 1958 with Luhn [12], who proposed analyzing word frequencies and distributions to compute the significance of sentences for summary creation. The increasing need for automatic document summarization drove more and more researchers into the area [1] [2] [3] [4] [11]. Document summarization can be categorized as either extractive or abstractive. The extractive approach selects sentences from the original document with little alterations to compose the summary. The abstractive approach utilizes new text which does not appear in the source. A system is referred to as generic summarization when its purpose is to capture the key meaning of input sources without special stress on any direction. By contrast, those producing summaries relevant to user queries are called query-based summarization.

In text summarization, a number of different approaches have been proposed to select the most relevant sentences. H. Takamura and M. Okumura [13] evaluate sentences according to cluster-based or graph-based models. The approach recently proposed in D. Wang and T. Li [14] exploits an incremental hierarchical clustering algorithm with the two-fold aim of identifying groups of sentences that share the same content and updating summaries over time. Lexrank [15] proposed to represent correlations among sentences by means of a graph-based model. Most relevant sentences are selected according to the eigenvector centrality computed by means of the well-known PageRank algorithm [12]. A parallel research effort has been devoted to formalizing the summarization task as a maximum coverage problem with Knapsack constraints based on sentence relevance within each document. However, previous approaches typically focus on single word significance which does not effectively capture correlations among multiple words at the same time. The paper [16], however, even though implementing only a very limited ontology, points at a promising direction in choosing sentences for summarization.

A number of researchers have made their summarization tools available on the Internet either for downloading and use in standalone mode, or direct use on the web. Despite that, commercial tools are also available. This paper proposes a new summarization strategy that assembles the results of six accurate Internet-based summarization tools to generate a single summary. This strategy also provides a comparative assessment between them. A new test corpus was developed with news from CNN (www.cnn.com). The advantage of the use of this new corpus rests on the very high quality of the

text and the highlights offered for each text, which is a good quality summary of 3 or 4 sentences. The CNN corpus encompasses 200 texts and is possibly one of the largest existing test *corpus* for summarization today.

2. THE SUMMARIZATION TOOLS

Six summarization tools were chosen to provide input to compose the "final" summary generated in the approach proposed here. They are: TextCompactor [5], FreeSummarizer [6], Smmry [7], WebSummaryzer [8], Interllexer [10], and Compendium [9]. A brief description of each of them is presented.

2.1 TextCompactor

TextCompactor is a free online summarization tool, created by Keith Edyburn for Knowledge by Design, Inc. It is used to help struggling readers process overwhelming amounts of information. In order to summarize the text, it calculates the frequency of each word in the passage. Then, a score is calculated for each sentence based on the frequency count associated with the words it contains. The most important sentence is deemed to be the sentence with the highest frequency count. The Text Compactor works best on expository text such as textbooks and reference material and it is not recommended for use with fiction (i.e., stories about imaginary people, places, and events).
The tool works on-line: the user submits a .txt file, and the output is in the same format. Sentences chosen are unchanged from the source file. TextCompactor cannot handle long input files (greater than 15,000 characters), and document structure is not taken into account.

2.2 Free Summarizer

Free Summarizer creates an extractive summary based on word frequencies. The service is free. It allows the user to select the number of sentences in the summary. Like TextCompactor, sentences chosen are not changed from the source file, long input files (greater than 15,000 characters) are not processed, and document structure is not taken into account.

2.3 Smmry

Smmry was created in 2009 by Amir Elmaani. It creates a summary following these five steps:1) Ranking sentences by importance using the core algorithm; 2) Reorganizing the summary to focus on a topic; by selection of a keyword; 3) Removing transition phrases; 4) Removing unnecessary clauses; 5) Removing excessive examples.
The core algorithm calculates the occurrence of each word in the text, after associating words with their grammatical counterparts. Then it ranks sentences by the sum of points of the words in it. The tool was developed in PHP, works on-line and as an API having as input either .txt or html files, and producing output of the same file type. The output sentences may be slightly modified as transition phrases, unnecessary appositives, and excessive examples are removed.

2.4 WebSummaryzer

WebSummarizer is an application designed by Context Discovery Inc. It supports summarization of content in English, French, German and Spanish. The summary is created using sentence rank, and it is present as a structured outline and a Visual Summary. The Visual Summary and the structured outline are interactive content maps that users can navigate in their browsers by keywords to instantly see the key summaries in context. It is important to note that none of the summaries are created or reviewed by people; the entire process is automatic. There is an on-line trial version and an API for licensing. This tool handles input of several kinds: plain text, e-mails, doc, pdf, etc. The output may be in .txt. The chosen sentences are kept unchanged. The tool is able to handle large files (of size greater than 15,000 characters). The document type information is not used.

2.5 Interllexer Summarizer

The commercial tool called Intellexer Document Summarizer [10] is a desktop application in two different versions: one general purpose and another professional (Pro). While both the versions have the same user interface and an identical set of functions, the difference between them lies in the internal algorithms of operation and vocabulary packs.
The Professional version has the following features:

- claims to offer professional quality of summarizing even for text as complicated as documents for lawyers, researchers or news analysts.

- suitable for special-purpose documents: such as patents, scientific articles, economic reviews, etc.
- tuned for subjects in: General, Patent, Scientific, Economics, Politics, Law, Health, ITechnology, Disaster, Ecology, Sports, Innovation.
- Compatible with files PDF, TXT, HTML/HTM, DOC, PPT, RTF, CHM, URL, DOCX, MHTML/MHT.
- Has neither input nor output limits.

A 30-day demo version is available from the product site: http://summarizer.intellexer.com/summ_demo_v2.php.

2.6 Compendium

Compendium [2] is a text summarization tool capable of generating the most common types of summaries. With this tool the user can generate extractive and abstractive summaries from a single or multiple documents, either query-focused or sentiment-based. The main contributions of compendium are the:

1) use of textual entailment for avoiding redundant information in the summaries;
2) combination of statistical and cognitive-based techniques for detecting relevant information; and
3) generation of abstractive-oriented summaries.

Compendium was developed by Elena Lloret and Manuel Palomar and performs:

1) a surface linguistic analysis (tokenization, POS-tagging, stemming, stop word identification),
2) redundancy detection (textual entailment (TE) as a technique to detect redundancy),
3) topic identification (identify the main topics of a document),
4) relevance detection (the relevance detection stage assigns a weight to each sentence, depending on how relevant it is within the text, using The Code Quantity Principle),
5) summary generation (the most important sentences, i.e. the ones with the highest scores, are selected and extracted).

The tool works on-line, and takes a .txt file as input and generates another .txt file as output. The chosen sentences are not modified. Although it is not able to handle files longer than 15,000 characters, it is able to receive several documents as input for summarization. Document structure is not taken into account.

3. AN EXAMPLE OF SUMMARIZATION

This section describes the strategy adopted for the generation of a new compound summary, having as input the output of the tools outlined in the last section. The basic strategy is to compare the frequency of sentences in the output of the summaries and to choose the most frequent ones to be part of the output—sentence-based voting. To exemplify the result of the scheme presented the webpage transcribed below with sentences numbered:

[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National Highway Traffic Safety Administration.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[5] That reduced clearance may mean the driver's foot could brush the side of the brake pedal when going from the accelerator to the brake, thus increasing stopping distances and the risk of a crash.
[6] Gas prices still slipping, survey says
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.

The summary provided by Compendium [2] summarizer for the text above is:

[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[7] Gas prices still slipping, survey says Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.

FreeSummarizer and Smmary yield as result:

[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.

Interllexer results are:

[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[5] The reduced clearance may mean the driver's foot could brush the side of the brake pedal when going from the accelerator to the brake increasing stopping distances and the risk of a crash.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.

TextCompactor yielded the following summary:

[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[5] That reduced clearance may mean the driver's foot could brush the side of the brake pedal when going from the accelerator to the brake, thus increasing stopping distances and the risk of a crash.
[6] Gas prices still slipping, survey says

Finally, the WebSummarizer, with the set-up short, provided the following output:

[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National Highway Traffic Safety Administration.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[7] Gas prices still slipping, survey says Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.
[8] The automaker will notify owners, the administration said, but customers may also contact the NHTSA's vehicle safety hotline at 1-888-327-4236 or go to www.safercar.com.

4. PRELIMINARY RESULTS

The 4-sentence summary obtained using the voting-based scheme presented here is:

[1] The Ford Motor Company is recalling more than 8,000 of its 2013 Escape compact SUVs because of a potential problem affecting the brake pedal.
[3] There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.

The summary above appears to be a representative account of the original text. The CNN article analyzed has a number of virtues: it is extremely concise, clear, and objective. Besides that, it offers author-provided text highlights. The highlights of the text presented from the CNN-site are:

- The recall affects 8,266 Escape SUVs.
- Mispositioned carpet can reduce clearance around the brake pedal.
- Dealers will correct the problem free of charge.

One can easily relate each of the sentences of the highlights above to the following sentences, respectively:

[2] The recall affects SUVs made between March 8 and June 7, 2012, according to the National Highway Traffic Safety Administration. There are 8,266 Escapes involved in the recall.
[4] Ford says mispositioned carpet padding on the center console trim panel may be pushed outward, reducing clearance with the pedals, according to the NHTSA.
[7] Ford dealers will remove the carpet padding and replace the left-side console trim panel free of charge, the NHTSA said.

A qualitative assessment of large data sets in a difficult task; thus, researchers look for quantitative assessment methods for summaries. ROUGE [17] is one of the tools most widely used for such purpose. Having the sentences in the highlight as gold standard provides the following results (95% confidence interval – parameters -e ../data -c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a):

Table 1. Results of ROUGE for the summaries presented having the highlights as gold standard.

	Compend.	FreeS	Interllexer	Summry	TextCom	WebSum	Proposed
Average_R	0.54930	0.56338	0.53521	0.56338	0.47887	0.78873	0.74648
Average_P	0.30709	0.22857	0.24516	0.22857	0.28099	0.43750	0.38971
Average_F	0.39394	0.32520	0.33628	0.32520	0.35416	0.56281	0.51208

The calculus of ROUGE having as gold standard the sentences in the text that best match the highlights yielded exactly the same results as the one of having the CNN-highlights as gold standard for this text. This is not always the case, as demonstrated below.

5. GENERAL RESULTS

To better assess the results of summarization using the tools presented and the strategy proposed, a CNN corpus was developed with 250 texts evenly assigned to four categories: technology, travel, sports, business, news of the world, Latin America, Europe, and Middle-East. The frequency of coincident results per text nature may be seen in Table 2.

If one chooses for the compound summary sentences that were chosen by at least three classifiers one would have an average of 3.74 sentences per summary according to the data presented in Table 2. As that is an average number, a minimum of four sentences and a maximum of six were chosen for the compound summary. The strategy adopted to either discard sentences or provide the minimum number was:

1. Calculate the ROUGE of each summarizer using the CNN corpus.
2. In the case of having to discard sentences (there are more than 6 sentences in the summary), for each sentence calculate its weight as being the sum of the Average_R score of each of the summarizers that chose that sentence. The sentences with the lowest scores are discarded.
3. In the case of having to complete the summary to get a minimum of 4 sentences, the strategy adopted borrows sentences from the summarizer with the highest ROUGE score, but the sentences chosen obey an even spacing sentence number distribution, to better represent the whole text.

Table 2. Coincidence frequency of sentences chosen by summarization tools.

Subject	F=6	F=5	F=4	F=3	F=2	F=1	# Sentences	# Texts
Technology	11	14	24	46	97	411	817	25
Travel	5	11	21	29	45	126	1,625	25
Sports	7	16	45	43	68	171	635	25
Business	7	16	32	47	62	136	457	25
World_News	4	11	10	0	0	0	613	25
Latin_America	8	12	5	0	0	0	413	25
Europe	24	1	0	0	0	0	674	25
Middle_East	35	25	15	0	0	0	1,700	75
Total	**101**	**106**	**152**	**165**	**272**	**844**	**6,934**	**250**

As an example of how these rules were implemented, suppose a compound summary was formed with two sentences: [17] (F=3) and [26] (F=2). The summarizer with the highest ROUGE score selected 5 sentences: [3] [7] [17] [25] [32]. The compound summary must have a minimum of 4 sentences and [17] and [26] were already chosen by the global strategy, then one has to choose two sentences out of: [3], [7], [25] and [32]. Taking [17] as a reference the distances are: 14, 10, 8, 15. From [26] the distances are: 23, 19, 1, 6. Sentence number [3] has the largest distance from the set, thus it is included in the compound summary. Now, the new distances are: from [3]: 4, 22, 29; from [17]: 10, 8, 15; from [26]: 19, 1, 6. The largest global distance is of sentence [32]. Thus, the final MECKS summary has sentences: [3], [17], [26], [32]. Such a strategy of choosing sentences is related to k-means calculation.

The result of calculating ROUGE for the summarizers presented and the compound summarizer for the 250 texts in the CNN corpus, using the highlights of the CNN articles as the gold standards, is shown in Table 3. The results for having the sentences that more closely match the highlights being used as the gold standards are shown in Table 4.

Table 3. Results of ROUGE having the highlights as gold standard.

	Compend.	FreeS	Interllexer	Proposed	Summry	TextCom	WebSum
Average_R	0.51 ±0.16	0.51 ±0.13	0.52 ±0.16	0.56 ±0.15	0.55 ±0.15	0.58 ±0.17	0.52 ±0.14
Average_P	0.18 ±0.07	0.18 ±0.07	0.17 ±0.06	0.19 ±0.07	0.16 ±0.06	0.16 ±0.07	0.19 ±0.06
Average_F	0.26 ±0.09	0.25 ±0.07	0.27 ±0.09	0.25 ±0.08	0.24 ±0.08	0.26 ±0.08	0.27 ±0.08

If one compares the ROUGE scores, the TextCompactor would provide the best summaries and the compound summary (Proposed) would provide the second best one.

Table 4. Results of ROUGE having the sentences that match the highlights as gold standard.							
	Compend.	**FreeS**	**Interllexer**	**Proposed**	**Summry**	**TextCom**	**WebSum**
Average_R	0.55 ±0.20	0.56 ±0.18	0.58 ±0.21	0.65 ±0.21	0.62 ±0.20	0.65 ±0.23	0.57 ±0.20
Average_P	0.42 ±0.18	0.42 ±0.19	0.39 ±0.13	0.46 ±0.18	0.39 ±0.16	0.38 ±0.18	0.44 ±0.17
Average_F	0.45 ±0.15	0.46 ±0.15	0.45 ±0.13	0.52±0.16	0.46±0.16	0.45 ±0.16	0.49 ±0.17

The relative results did not change when the ROUGE was calculated having the sentences that best match the highlights as gold standard.

CONCLUSIONS

This paper presents a new strategy for text summarization taking several summarization tools as input and composing the results to yield a better summary. The strategy proposed seems highly promising in terms of reaching a better summarization standard. A high-quality corpus using news articles of CNN was developed for setting fair comparision grounds. This work also presents a systematic assessment of six of the best summarization tools available. The quantitative assessment made using ROUGE identified TextCompactor [5] as the best of the six tools tested for these text samples, and the compound summary proposed herein as having the second best quantitative results.
The authors are currently enlarging the CNN-corpus to 400 texts and developing a qualitative assessment of the quality of each tool and the compound summarization strategy. Further tests are being conducted with the DUC [17] corpus.

ACKNOWLEDGMENTS

This research was partly sponsored by the National Council for Scientific and Technological Development (CNPq/Brazil) and by HP Brazil Ltda., using resources coming from tax exemption provided by Brazilian Law 8.248/1991 (Informatics Law).

REFERENCES

[1] Ani Nenkova and Kathleen McKeown. A Survey of Text Summarization Techniques. http://www.springerlink.com/content/m136272x2862nx51/?MUD=MP
[2] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review http://www.springerlink.com/content/4455125331140684/
[3] Carlos Méndez Cruz and Alfonso Medina Urrea. Extractive Summarization Based on Word Information and Sentence Position http://www.springerlink.com/content/xt7f5bdwpgduve43/
[4] Xiaoyue Liu, Jonathan J. Webster and Chunyu Kit. An Extractive Text Summarizer Based on Significant Words. http://www.springerlink.com/content/wm81647211u3kh67/
[5] TextCompactor http://www.textcompactor.com . Last visited 06/08/2012.
[6] FreeSummarizer Last visited 06/08/2012.
[7] Smmry. http://smmry.com. Last visited 06/08/2012.
[8] WebSummarizer http://websummarizer.com. Last visited 06/08/2008.
[9] Interllexer http://summarizer.intellexer.com/index.html. Last visited 14/08/2012.
[10] Sparck-Jones,K.:Automatic summarising: Factors and directions. In: Mani,I., Maybury,M. (eds.) Advances in Automatic Text Summarization, pp.1–12. MIT Press, London (1999).
[11] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. 7th WWW Conference, pp 107–117, 1998.
[12] H. Takamura and M. Okumura. Text summarization model based on the budgeted median problem. In 18th ACM Conference on Information and knowledge management problem. 1589–1592, 2009.
[13] D. Wang and T. Li. Document update summarization using incremental hierarchical clustering. Proc. of the 19th ACM international conference on Information and knowledge management, pp. 279-288, 2010.
[14] Erkan, G. and Radev, D.R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. J Artif Intell Res (JAIR) 22:457–479, 2004.
[15] Hennig, L., Wetzker, R., and Umbrath, W. "An Ontology-Based Approach to Text Summarization". IEEE/WIC/ACM Intern. Conference on Web Intelligence and Intelligent Agent Technology, 2008.
[16] Lin, CY. ROUGE: a package for automatic evaluation of summaries. ACL text summarization workshop. pp 74–81, 2004.
[17] Document Understanding Conference. HTL/NAACL workshop on text summarization, 2004.

Expert Systems with Applications 40 (2013) 5755–5764

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Assessing sentence scoring techniques for extractive text summarization

Rafael Ferreira [a,*], Luciano de Souza Cabral [a], Rafael Dueire Lins [a], Gabriel Pereira e Silva [a], Fred Freitas [a], George D.C. Cavalcanti [a], Rinaldo Lima [a], Steven J. Simske [b], Luciano Favaro [c]

[a] Informatics Center, Federal University of Pernambuco, Recife, Brazil
[b] Hewlett–Packard Labs., Fort Collins, CO 80528, USA
[c] Hewlett–Packard Brazil, Barueri, Brazil

ARTICLE INFO

Keywords:
Extractive summarization
Sentence scoring methods
Summarization evaluation

ABSTRACT

Text summarization is the process of automatically creating a shorter version of one or more text documents. It is an important way of finding relevant information in large text libraries or in the Internet. Essentially, text summarization techniques are classified as Extractive and Abstractive. Extractive techniques perform text summarization by selecting sentences of documents according to some criteria. Abstractive summaries attempt to improve the coherence among sentences by eliminating redundancies and clarifying the contest of sentences. In terms of extractive summarization, sentence scoring is the technique most used for extractive text summarization. This paper describes and performs a quantitative and qualitative assessment of 15 algorithms for sentence scoring available in the literature. Three different datasets (News, Blogs and Article contexts) were evaluated. In addition, directions to improve the sentence extraction results obtained are suggested.

1. Introduction

The rapid growth of the Internet yielded a massive increase of the amount of information available, especially regarding text documents (e.g. news articles, electronic books, scientific papers, blogs, etc.). Due to the huge volume of information in the Internet, it has become unfeasible to efficiently sieve useful information from the huge mass of documents. Thus, it is necessary to use automatic methods to "understand", index, classify and present all information in a clear and concise way, allowing users to save time and resources.

One solution is use text summarization techniques. Text summarization (TS) is the process of automatically creating a compressed version of one or more documents. It attempts to get the "meaning" of documents. Essentially, TS techniques are classified as *Extractive* and *Abstractive* (Lloret & Palomar, 2012). Extractive summaries produce a set of the most significant sentences from a document, exactly as they appear. Abstractive summaries attempt to improve the coherence among sentences by eliminating redundancies and clarifying the contest of sentences. It may even produce new sentences to the summary. Currently, the extractive summaries are commonly used because they are easier to create. Due to this in this work we focus on them.

Extractive methods are usually performed in three steps (Nenkova & McKeown, 2012):

- Create an intermediate representation of the original text;
- Sentence scoring;
- Select high scores sentences to the summary.

The first step creates a representation of the document. Usually, it divides the text into paragraphs, sentences, and tokens. Sometimes some preprocessing, such as stop word removal is also performed. The second step tries to determine which sentences are important to the document or to which extent it combines information about different topics, by sentence scoring. The score should be a measure of how significant a sentence is to the "understanding" of the text as a whole. The last step combines the score provided by the previous steps and generates a summary.

This paper describes 15 sentence scoring methods, and some variation of them, widely used and referenced in the literature applied to single document summarization in the last 10 years. Ani Nenkova points in Nenkova and McKeown (2012) three other types of sentence selection: bayesian topic models, sentence clustering, and domain-dependent topics. These methods are not explored in this paper, because the results yielded are not considered up to the same level as the others do not yet (Nenkova & McKeown, 2012). Each of the 15 scoring methods is described and implemented.

* Corresponding author. Tel.: +55 8197885665.
E-mail addresses: rflm@cin.ufpe.br (Rafael Ferreira), lscabral@gmail.com (L. de Souza Cabral), rdl@cin.ufpe.br (R.D. Lins), gfps.cin@gmail.com (G. Pereira e Silva), fred@cin.ufpe.br (F. Freitas), gdcc@cin.ufpe.br (G.D.C Cavalcanti), rjlima01@gmail.com (R. Lima), steven.simske@hp.com (S.J. Simske), luciano.favaro@hp.com (L. Favaro).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.04.023

A quantitative and qualitative assessment of those methods using three different datasets (news, blogs, and articles context) is performed. The precision and recall measures (Baeza-Yates & Ribeiro-Neto, 1999) provided by ROUGE (Lin, 2004) were used to perform the quantitative assessment of the studied methods. The qualitative assessment was performed by four people who analyzed each original text and selected the sentences that they feel ought to be in the summary. The qualitative evaluation is done by counting the numbers of sentences selected by the system that match the human gold standard. Processing-time performance of each of the algorithm implemented is also taken into account.

It is important to notice that Lloret and Palomar (2012) and Nenkova and McKeown (2012) present two recent and comprehensive surveys on text summarization. They do not present any assessment of any sort of the techniques and this paper targets at filling in such an important gap.

In addition, some directions on "How Can Sentence Scoring Results be Improved?" are presented. Orasan (2009) and Nenkova and McKeown (2011) point that the main directions to do it are:

- Morphological transformation;
- There is often a large amount of words with little meaning to the text (stop words);
- The use of synonyms, words with similar semantics, may obscure the "weight" of a given word in the text in frequency-based methods;
- Co-reference;
- Ambiguity; and
- Redundancy.

This paper is structured as follows Section 2 presents the algorithms for sentence scoring more used in the technical literature. Section 3 explains the assessment parameters used. Section 4 presents the results of the quantitative, qualitative assessment of the algorithms together with the measures of time performance. Section 5 describes some problems that affect the results of the algorithms studied. In the conclusions, an account of the contribution made is presented together with lines for further work.

2. Sentence scoring methods

The first reference to text summarization using sentence scoring dates back to 1958 (Luhn, 1958; Lloret & Palomar, 2009). As already stated, the focus of these research areas are addressed by the following question: how can a system determine which sentences are representative of the content of a given text? In general, three approaches are followed: (i) *Word scoring* – assigning scores to the most important words; and (ii) *Sentence scoring* – verifying sentences features such as its position in the document, similarity to the title, etc.; and (iii) *Graph scoring* – analyzing the relationship between sentences.

The following section presents the main methods in each of the aforementioned approaches.

2.1. Word scoring

The initial methods in sentence scoring were based on words. Each word receives a score and the weight of each sentence is the sum of all scores of its constituent words. The approaches in the literature are outlined here.

2.1.1. Word frequency

As the name of the method suggests, the more frequently a words occurs in the text, the higher its score (Luhn, 1958; Lloret & Palomar, 2009; Gupta et al., 2011; Kulkarni & Prasad, 2010; Abuobieda, Salim, Albaham, Osman, & Kumar, 2012). In other words, sentences containing the most frequent words in a document stand a higher chance of being selected for the final summary. The assumption is that the higher the frequency of a word in the text, the more likely that it indicates the subject of the text.

2.1.2. TF/IDF

The hypothesis assumed by this approach is that if there are "more specific words" in a given sentence, then the sentence is relatively more important. The target words are usually nouns except for temporal or adverbial nouns (Satoshi et al., 2001; Murdock, 2006). This algorithm performs a comparison between the term frequency (*tf*) in a document (in this case each sentence is treated as a document) and the document frequency (*df*), which means the number of times that the word occurs along all documents. The *TF/IDF* score is calculated as follows:

$$TF/IDF(w) = DN\left(\frac{\log(1+tf)}{\log(df)}\right) \quad (1)$$

where *DN* is the number of documents.

2.1.3. Upper case

This method assigns higher scores to words that contain one or more upper case letters (Prasad et al., 2012). It can be a proper name, initials, highlighted words, among others. The score is calculated as follows:

$$CPTW(j) = \frac{NCW(j)}{NTW(j)} \quad (2)$$

where:

CPTW = Ratio of total first letter capital words present in the sentence to the total number of words present in the sentence,
NCW = Number of first letter capital words, and
NTW = Total number of words present in sentence.

$$UCf = \frac{CPTW(j)}{MAX(CPTW(j))} \quad (3)$$

where, *UCf* = Uppercase feature value.

2.1.4. Proper noun

Usually the sentences that contain a higher number of proper nouns are more important; thus, they are likely to be included in the document summary (Fattah & Ren, 2009). This is a specialization of the *Upper case* method.

2.1.5. Word co-occurrence

Word co-occurrence measures the chance of two terms from a text appear alongside each other in a certain order. One way to implement this measure is using *n*-gram (Mariño et al., 2006), which is a contiguous sequence of n items from a given sequence of text or speech. In short, it gives higher scores to sentences that co-occurrence words appear more often (Liu, Webster, & Kit, 2009; Gupta et al., 2011; Tonelli & Pianta, 2011).

2.1.6. Lexical similarity

It is based on the assumption that important sentences are identified by strong chains (Gupta et al., 2011; Barrera & Verma, 2012; Murdock, 2006). In other words, it relates sentences that employ words with the same meaning (synonyms) or other semantic relation.

2.2. Sentence scoring

This approach analyzes the features of the sentence itself and was used for the first time in 1968 (Edmundson, 1969) analyzing the presence of cue words in sentences. The main approaches that follow this idea are described below.

2.2.1. Cue-phrases

In general, the sentences started by "in summary", "in conclusion", "our investigation", "the paper describes" and emphasizes such as "the best", "the most important", "according to the study", "significantly", "important", "in particular", "hardly", "impossible" as well as domain-specific bonus phrases terms can be good indicators of significant content of a text document (Gupta et al., 2011; Kulkarni & Prasad, 2010; Prasad et al., 2012). A higher score is assigned to sentences that contain cue words/phrases, using the formula:

$$CP = \frac{CPS}{CPD} \tag{4}$$

where,

CP = Cue-phrase score,
CPS = Number of cue-phrases in the sentence,
CPD = Total number of cue-phrases in the document.

2.2.2. Sentence inclusion of numerical data

Usually the sentence that contains numerical data is an important one and it is very likely to be included in the document summary, according to references (Fattah & Ren, 2009; Kulkarni & Prasad, 2010; Abuobieda et al., 2012; Prasad et al., 2012). This kind of sentence usually refers to some important information such as date of event, money transaction, damage percentage, etc.

2.2.3. Sentence length

This feature is employed to penalize sentences that are too short (Fattah & Ren, 2009) or too long (Abuobieda et al., 2012), these sentences are not considered as an optimal selection. The method uses length as number of words in sentence. In addition, Satoshi et al. (2001) penalizes sentences that are shorter than a certain length.

The first case could be calculated as follows:

$$Score = Length(s) * AverageSentenceLength \tag{5}$$

The penalty score is calculated using a conditional:

$$Score(Si) = \begin{cases} Li & \text{if } (Li > C) \\ Li - C & otherwise \end{cases} \tag{6}$$

where,

Li = length of sentence i and
C = certain length defined by user.

2.2.4. Sentence position

There are many approaches that use the sentence position as a score criterion (Fattah & Ren, 2009; Satoshi et al., 2001; Barrera & Verma, 2012; Abuobieda et al., 2012; Gupta et al., 2011). In reference (Abuobieda et al., 2012), the first sentence in the paragraph is considered an important sentence and a strong candidate to be included in the summary; Gupta et al. (2011) says that the first sentences of paragraphs and words in titles and headings are more relevant to summarization; The method proposed in reference (Satoshi et al., 2001) assigns score 1 to the first N sentences and 0 to the others, where N is a given threshold for the number of sentences.

Fattah and Ren (2009) follow the same principle as reference (Satoshi et al., 2001) and assume that the first sentences of a paragraph are the most important ones. The sentences are ranked as follows: the first sentence in a paragraph has a score value of 5/5, the second sentence has a score 4/5, and so on. Sentences further embedded in the paragraph are not significant. The latest approach in the literature (Barrera & Verma, 2012) exploits three position models. The first assumes that sentences closer to the start and end of a document are more likely to be more content representative. The second prioritizes only the top parts of the text. The last one uses sentences close to topic headings to create the summary.

2.2.5. Sentence centrality

Sentence centrality is the vocabulary overlap between a sentence and other sentences in the document (Fattah & Ren, 2009; Abuobieda et al., 2012; Kulkarni & Prasad, 2010). This approach makes no use any semantic treatment as Lexical Similarity. Another way to treat this measure is using other sentence similarity algorithms, for example, Bleu (Haque, Naskar, Way, Costa-jussa, & Banchs, 2010). Centrality could be calculated as follows:

$$Score = \frac{Ks \cap KOs}{Ks \cup KOs} \tag{7}$$

where,

Ks = Keywords in s, and
KOs = Keywords in other sentences.

2.2.6. Sentence resemblance to the title

Sentence resemblance to the title is the vocabulary overlap between this sentence and the document title (Satoshi et al., 2001; Fattah & Ren, 2009; Kulkarni & Prasad, 2010; Abuobieda et al., 2012). In this case, sentences similar to the title and sentences that include the words in the title are considered important. A simple way to calculate this score is:

$$Score = \frac{Ntw}{T} \tag{8}$$

where,

Ntw = Number of title words in sentence, and
T = Number of words in the title.

2.3. Graph scoring

In graph-based methods the score is generated by the relationship among the sentences. When a sentence refers to another it generates a link with an associated weight between them. The weights are used to generate the score of sentences.

2.3.1. Text rank

TextRank is a graph-based ranking model for text processing (Barrera & Verma, 2012; Mihalcea & Tarau, 2004). It extracts important keywords from a text document and also to determine the weight of the "importance" of words within the entire document by using a graph-based model. Sentences with a larger quantity of keywords get higher scores.

2.3.2. Bushy path of the node

The bushy path of a node (sentence) on a map is defined as the number of links connecting it to other nodes (sentences) on the map (Fattah & Ren, 2009).

2.3.3. Aggregate similarity

Aggregate similarity measures the importance of a sentence. Instead of counting the number of links connecting a node (sentence)

to other nodes (Bushy Path), aggregate similarity sums the weights (similarities) on the links (Fattah & Ren, 2009).

3. Evaluation parameters

This section describes: (i) the datasets used; (ii) methodology followed in the experiments to assess the quality of summaries; (iii) the computer used to perform the experiments.

3.1. Corpus

Three different datasets were used for testing the performance of the scoring methods presented. They are detailed in the following subsections.

3.1.1. CNN Dataset

The CNN corpus developed by Lins and colleagues (Lins et al., 2012) encompasses news articles from all over the world. The current version of this corpus presents 400 texts assigned to 11 categories: Africa, Asia, business, Europe, Latin America, Middle East, US, sports, tech, travel, and world news. The texts were selected from the news articles of CNN website (http://www.cnn.com). Besides the very high quality, conciseness, general interest, up-to-date subject, clarity, and linguistic correctness, one of the advantages of this new corpus is that a good-quality summary for each text, called the "highlights, is also provided. The highlights are three or four sentences long and are of paramount importance for evaluation purposes, as they may be taken as a summary of reference, or gold standard. In addition, two new summary evaluation sets were created. The first one was obtained by mapping the sentences in the highlights onto the original sentences of the text. The second one was generated by the authors blindly reading the texts and selecting n sentences that one thought better described each text. The value of n was chosen depending on the text size, but in general it was equal to the number of sentences in the highlight plus two. The most voted sentences were chosen and a very high sentence selection coincidence was observed. This second test set encompassed the first one in all cases.

3.1.2. Blog summarization dataset

In 2008, Hu and colleagues (Hu, Sun, & Lim, 2007, 2008) felt the need to get a Blog benchmark dataset. Thus, they decided to collect data from two blogs, Cosmic Variance (http://cosmicvariance.com) and Internet Explorer Blog (http://jblogs.msdn.com/i.e./). Both have large numbers of posts and comments. From those blogs, 100 posts, 50 from each blog, were randomly chosen to form the evaluation dataset. To generate reference summaries, four human summarizers read the all chosen posts and their corresponding comments and then selected approximately seven sentences from each post.

3.1.3. SUMMAC dataset

The SUMMAC Corpus was elaborated under responsibility of the MITRE Corporation in cooperation with the University of Edinburgh, as part of the SUMMAC conference organizer group (Tipster Text Summarization Evaluation Conference) effort.[1] This dataset has 183 papers on Computation and Language, obtained from the repository LANL (Los Alamos National Laboratory) maintained by Cornell University Library, which currently holds more than 800,000 electronic documents from various fields in their database. After selecting the documents, they were annotated in XML, taking up their sections identified, being made available to Information Recovery, Extraction and Summarization. Information can be obtained through the link: http://www-nlpir.nist.gov/related_projects/tipster_summac/cmplg-xml.tar.gz.

[1] http://www-nlpir.nist.gov/related_projects/tipster_summac/cmp_lg.html.

3.2. Evaluation methodology

This section describes the methodology followed in the experiments to assess the quality of summaries.

3.2.1. Quantitative assessment

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) was used to quantitatively evaluate the summaries generated by using the different scoring methods. ROUGE is widely used for such purpose. This fully automated evaluator essentially measures the content similarity between system-developed summaries and the corresponding gold summaries.

The result of calculating ROUGE for the CNN dataset summaries is presented in two perspectives: (i) using the highlights of the CNN articles as the gold standards; and (ii) using the sentences that more closely match the highlights as the gold standards.

In relation to the blog summarization dataset all summaries (this dataset contains four summaries as presented in Section 3.1.2) are used as ROUGE input. At last, the SUMMAC Dataset provides the article abstract as input to ROUGE.

3.2.2. Qualitative assessment

The qualitative evaluation was performed in the CNN and Blog Summarization Dataset Corpora. As mentioned before, four people analyzed each original text and selected the sentences that they feel ought to be in the summary of those datasets. The qualitative evaluation is done by counting the numbers of sentences selected by the system that match the human gold standard. The SUMMAC Dataset provides only an abstract, which is not adequate to the assessment performed here.

3.3. Computer specification

To perform the experiment we use a computer with following specification:

- *Operational system*: Windows 7 64 Bits;
- *Processor*: Intel (R) Core (TM) i7-2670, 2.20 GHz;
- RAM memory: 8 GB.

4. Summarization performance evaluation

This section presents: (i) some abbreviations to better understand the experimentation; and (ii) details about the implementation of each method; and (iii) the results of the evaluation of the performance of the algorithms. The assessment was performed using each dataset separately.

4.1. Abbreviations

In order to facilitate presentation of the results, Table 1 lists the abbreviations to the terms used in next section and Table 2 shows a set of abbreviations for the name the algorithms.

4.2. Implementation of the algorithms

All algorithms described in Section 2 were implemented as described:

Word frequency: This algorithm is divided into four steps: (i) Remove all stop words; (ii) Count the number of each word from text. This step creates a structure that connects the word to the number of times it appears in the text (Word Frequency

Table 1
Abbreviations.

Average_R	Recall average
Average_P	Precision average
Average_F	F-measure average
Alg	Algorithm

Score); (iii) For each sentence it adds up the word frequency score of each word in a sentence.

TF/IDF: This algorithm is divided into the following steps: (i) Remove all stop words; (ii) Calculate the formula presented in Section 2.1.2 (TF/IDF Score) for each word from text; (iii) For each sentence it sum the TF/IDF score of each word in sentence.

TF/IDF: It is divided into: (i) Remove all stop words; (ii) Count the number of words with capital letters in text; (iii) Calculate the formula presented in Section 2.1.3 (Upper Case Score).

TF/IDF: The processing in this algorithm is performed as: (i) Remove all stop words; (ii) Perform POS tagging (using Stanford CoreNLP[2]) in order to select only nouns; (iii) Count the number of nouns that starts with capital letters (Proper Noun Score); (iv) For each sentence, add up the proper noun score of each word in a sentence.

Word co-occurrence: It is divided into: (i) Compute *n*-gram measure to n = 2, 3 and 4. (ii) For each sentence, add up the *n*-gram score of each word in a sentence.

Lexical similarity: It uses WordNet[3] to find similarity among words than applies Word Frequency algorithm.

Cue-phrases: There are three steps to perform this algorithm: (i) Load a cue-phrase list[4]; (ii) Count the total number of cue-phrases in the document; (iii) Calculate the formula presented in Section 2.2.1 (Cue-phrases Score) for each sentence from text.

Sentence inclusion of numerical data: This algorithm uses regular expressions to verify if some numerical data is present in sentences.

Sentence length: It works as follows: (i) Calculate the largest sentence length; (ii) Penalize sentences larger than 80 percent of the largest sentence length; (iii) Calculate the Sentence Length Score for all other sentences.

Sentence position 1 and 2: This algorithm combines the position score presented in Fattah and Ren (2009) and Barrera and Verma (2012). In short, the sentences are ranked as follows: the first sentence in a text has a score value of 5/5, the second sentence has a score 4/5, and so on. The same thing occurs with the last sentences: the last one receives score value of 5/5, penultimate has a score 4/5, and so on. The sentence position 1 applies this concept considering all text. On the other hand, sentence position applies to each paragraph from text.

Sentence centrality 1: It uses Bleu measure (from HultigLib[5]) in order to verify the similarity among sentences.

Sentence centrality 2: It implements the formula presented in Section 2.2.5.

Resemblance to the title: This algorithm implements the formula presented in Section 2.2.6.

Aggregate similarity: It follows two steps: (i) to create the link among sentences using the sum of all measures (from HultigLib); (ii) to sum all links score for each sentence.

TextRank Score: It uses the textrank algorithm provided in https://github.com/turian/textrank.

[2] http://nlp.stanford.edu/software/corenlp.shtml.
[3] http://wordnet.princeton.edu/.
[4] http://www.cs.otago.ac.nz/staffpriv/alik/papers/apps.pdf.
[5] http://www.di.ubi.pt/~paulo/hultiglib/.

Table 2
Algorithms.

alg01	Word frequency
alg02	TF/IDF
alg03	Upper case
alg04	Proper noun
alg05	Word co-occurrence
alg06	Lexical similarity
alg07	Cue-phrase
alg08	Inclusion of numerical data
alg09	Sentence length
alg10	Sentence position 1
alg11	Sentence position 2
alg12	Sentence centrality 1
alg13	Sentence centrality 2
alg14	Resemblance to the title
alg15	Aggregate similarity
alg16	TextRank score
alg17	Bushy path

Bushy path: It is similar to aggregate similarity. Here, the algorithm counts the number of links differently from the previous one, which counts the link scores.

4.3. Assessment using CNN dataset

The result of calculating ROUGE for each algorithm, using CNN dataset as the gold standard, is shown in Table 3. Although the results obtained are close, some points should be remarked:

- Alg01, alg02 and alg09 achieved the best recall;
- Alg10, alg11, alg12 and alg14 reached the best precision;
- Alg01, alg02 and alg10 also achieved best *f*-measure;
- The Word scoring methods provided the best results of all the algorithms tested occupying the three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was alg02;
- The best sentence scoring algorithm was alg10;
- The best graph scoring algorithm was alg16.

Fig. 1 presents the results of qualitative evaluation. As mentioned before, it counts the number of sentences selected by the system that match the human gold standard. The highest scores were obtained by: alg02 (611), alg01 (601), alg14 (580), alg06 (570), and alg09 (553).

Time performance of each algorithm is presented in Table 4.

Table 3
Results of ROUGE having CNN dataset as gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
alg01	0.71(0.19)	0.35(0.13)	0.46(0.15)
alg02	0.73(0.17)	0.35(0.12)	0.46(0.15)
alg03	0.64(0.19)	0.35(0.12)	0.44(0.12)
alg04	0.64(0.20)	0.35(0.13)	0.45(0.15)
alg05	0.59(0.20)	0.33(0.13)	0.42(0.15)
alg06	0.69(0.19)	0.35(0.13)	0.46(0.14)
alg07	0.50(0.22)	0.35(0.13)	0.40(0.14)
alg08	0.56(0.21)	0.36(0.13)	0.43(0.14)
alg09	0.70(0.18)	0.33(0.12)	0.44(0.15)
alg10	0.61(0.22)	0.40(0.13)	0.47(0.15)
alg11	0.52(0.22)	0.36(0.13)	0.41(0.12)
alg12	0.46(0.25)	0.37(0.16)	0.38(0.15)
alg13	0.33(0.21)	0.31(0.13)	0.30(0.15)
alg14	0.67(0.20)	0.36(0.12)	0.46(0.14)
alg15	0.57(0.20)	0.34(0.12)	0.42(0.14)
alg16	0.62(0.20)	0.34(0.12)	0.43(0.14)
alg17	0.56(0.20)	0.35(0.13)	0.42(0.14)

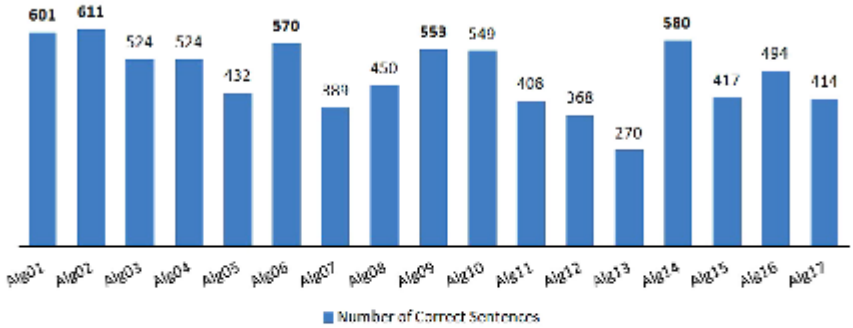

Fig. 1. Number of correct sentences x algorithms – using CNN dataset.

Table 4
Execution time using CNN dataset.

Alg	Execution time (s)
alg01	13.986
alg02	196.269
alg03	5.724
alg04	25.723
alg05	20.490
alg06	419.609
alg07	8.133
alg08	4.029
alg09	4.820
alg10	4.122
alg11	4.292
alg12	8.999
alg13	47.267
alg14	5.617
alg15	7.708
alg16	322.045
alg17	8.309

Table 5
Results of ROUGE having blog summarization dataset as the gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
alg01	0.72(0.13)	0.63(0.15)	0.67(0.14)
alg02	0.75(0.11)	0.63(0.15)	0.68(0.13)
alg03	0.58(0.16)	0.61(0.15)	0.59(0.15)
alg04	0.57(0.17)	0.63(0.14)	0.59(0.14)
alg05	0.65(0.14)	0.63(0.14)	0.63(0.13)
alg06	0.71(0.14)	0.63(0.14)	0.66(0.14)
alg07	0.52(0.18)	0.64(0.14)	0.57(0.15)
alg08	0.54(0.18)	0.63(0.15)	0.58(0.16)
alg09	0.76(0.11)	0.62(0.14)	0.68(0.13)
alg10	0.46(0.19)	0.60(0.13)	0.51(0.17)
alg11	0.52(0.18)	0.63(0.14)	0.56(0.16)
alg12	0.50(0.18)	0.65(0.14)	0.56(0.16)
alg13	0.46(0.20)	0.60(0.15)	0.51(0.18)
alg14	0.60(0.18)	0.64(0.13)	0.61(0.16)
alg15	0.58(0.18)	0.62(0.13)	0.59(0.16)
alg16	0.68(0.14)	0.63(0.14)	0.65(0.13)
alg17	0.58(0.17)	0.63(0.13)	0.60(0.15)

Some conclusions may be drawn from the results obtained:

- Combining the qualitative and quantitative assessments performed, one may conclude that the algorithms that yield better summarization are: alg01, alg02, alg14;
- The best result is obtained by alg02. It takes a longer time to execute than alg01 and alg14 (approximately 1,507 times longer than alg01 and 3,920 than alg14);
- Alg10 is the second fastest (behind only to alg08) and reaches a good quantitative results;
- Alg06 yields good results, but it is the slowest of all methods tested.

4.4. Assessment using the blog summarization dataset

The result of calculating ROUGE for each algorithm, using the blog summarization dataset, is shown in Table 5. Some points should be remarked:

- Alg01, alg02, alg06 and alg09 achieved the best results for recall;
- Alg12 and alg14 reached the best precision;
- Alg01, alg02 and alg09 also achieved the best f-measure;
- Again, the word scoring methods provided the best results of all the algorithms tested occupying three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was alg02;
- The best sentence scoring algorithm was alg09;
- The best graph scoring algorithm was alg16.

Fig. 2 presents the results of the qualitative evaluation. As already explained, this assessment counts the number of sentences selected by the system that match the human gold standard. The highest scores were obtained by: alg09 (563), alg06 (552), alg16(551), alg01 (545), and alg02 (537).

Table 6 presents the time elapsed in the execution of the different scoring algorithms for the blog summarization dataset.

Conclusions:

- Combining qualitative and quantitative evaluations the best algorithms are: Alg02, Alg06, and Alg09;
- The best summarization results is provided by Alg09 and it is the fastest in relation to algorithms that provide the best summarization results.
- Alg06 archives good results, but it is the slowest amongst the algorithms tested.

4.5. Using SUMMAC dataset

The result of calculating ROUGE for each scoring algorithm, using as input the SUMMAC dataset with the gold standard, is shown in Table 7. Some points are worth remarking:

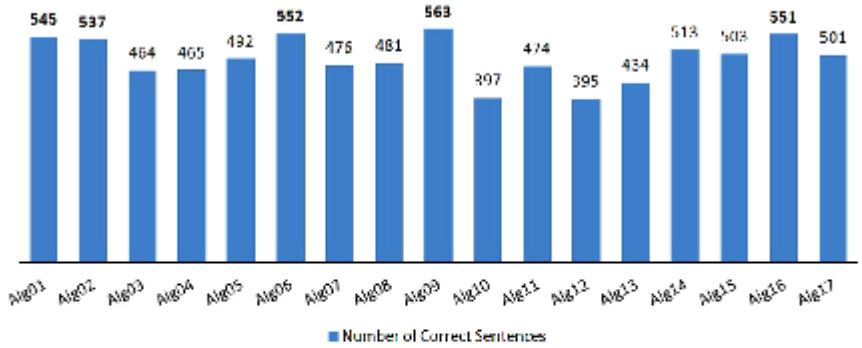

Fig. 2. Number of correct sentences x algorithms – using blog summarization dataset.

Table 6
Execution time using blog summarization dataset.

Alg	Execution time (s)
Alg01	2.508
Alg02	14.810
Alg03	1.841
Alg04	7.083
Alg05	2.943
Alg06	87.496
Alg07	2.982
Alg08	1.641
Alg09	2.391
Alg10	1.722
Alg11	1.799
Alg12	2.374
Alg13	5.225
Alg14	1.706
Alg15	2.194
Alg16	67.136
Alg17	2.508

Table 8
Execution time using SUMMAC dataset.

Alg	Execution time (s)
Alg01	17.287
Alg02	2,499.092
Alg03	9.606
Alg04	7.083
Alg05	73.948
Alg06	549.284
Alg07	29.954
Alg08	8.187
Alg09	9.670
Alg10	7.822
Alg11	7.750
Alg12	107.439
Alg13	2,602.518
Alg14	8.175
Alg15	38.316
Alg16	84.970
Alg17	39.903

Table 7
Results of ROUGE having SUMMAC dataset as gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
Alg01	0.48(0.10)	0.19(0.10)	0.26(0.10)
Alg02	0.47(0.11)	0.19(0.10)	0.26(0.10)
Alg03	0.25(0.11)	0.17(0.07)	0.19(0.06)
Alg04	0.22(0.11)	0.17(0.07)	0.18(0.06)
Alg05	0.23(0.16)	0.16(0.10)	0.17(0.10)
Alg06	0.46(0.11)	0.19(0.10)	0.26(0.10)
Alg07	0.33(0.11)	0.24(0.10)	0.26(0.07)
Alg08	0.25(0.11)	0.20(0.08)	0.21(0.07)
Alg09	0.49(0.09)	0.16(0.10)	0.23(0.09)
Alg10	0.31(0.10)	0.28(0.10)	0.28(0.06)
Alg11	0.24(0.11)	0.24(0.10)	0.22(0.08)
Alg12	0.07(0.11)	0.17(0.17)	0.07(0.08)
Alg13	0.22(0.11)	0.23(0.10)	0.21(0.08)
Alg14	0.36(0.14)	0.28(0.10)	0.29(0.08)
Alg15	0.22(0.08)	0.22(0.07)	0.21(0.05)
Alg16	0.46(0.10)	0.22(0.10)	0.28(0.09)
Alg17	0.23(0.10)	0.22(0.08)	0.21(0.06)

- Alg01, Alg02, and Alg09 achieved the best results for recall;
- Alg07, Alg08, Alg10, and Alg14 reached the best results for precision;
- Alg10, Alg14, and Alg16 also achieved the best *f*-measure;
- The sentence scoring methods provided the best results of all the algorithms tested occupying three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was Alg02;
- The best sentence scoring algorithm was Alg14;
- The best graph scoring algorithm was Alg16.

To conclude the experiments Table 8 presents the results of the time execution evaluation on SUMMAC dataset.

Conclusions:

- The best results were obtained by algorithms: Alg10, Alg14, and Alg16;
- Alg16 is in the top-3 in summarization performance, but it is the fifth slowest one;
- Alg10 is the third fastest (behind to Alg04 and Alg11) and it also archived good quantitative results;
- Alg02 yields good summarization results, but it is the second slowest one.

4.6. Discussion

Faced on the results presented in this section we draw the following conclusions.

Considering the CNN dataset the results are reasonable because the documents are better structured. In summary:

- The documents use well-formed words, therefore the alg01 and alg02 archive good results;

- Generally, in news texts important phrases are at the beginning and end of document, and they are concise. It explains the good results of alg10, alg11 and alg09;
- The alg14 archive good results because the journalists usually provide titles containing the main information of the news;
- alg12 has good precision because these kind of texts tend to be slightly redundant.
- alg06 archive good qualitative result because it uses synonymous to choose the sentence.

Differently, from experiment using CNN dataset, in Blog Summarization Dataset, the sentence-based algorithms do not archive good performance, it is caused because in this type of text the writers are not concerned about the structure of text. Thus, algorithms like alg10 did not get result as good as in previous experiment.

The alg01, alg02, and alg06, once again, reached good results. It happens because, in general, social web tools (like blogs) are based in topic words. In summary, significant words in this kind of text are important.

Alg14 and alg09 have good results because blogs usually have small text, which implies: (i) that the title characterizes the text, and (ii) that the sentences are, in general, lower.

As happened all experiments, in the evaluation using SUMMAC dataset, the alg01 and alg02 archive good recall. The difference here is the alg09. It probably archive these recall because the authors usually use concise sentences to express main ideas.

The precision was higher in three sentence-based algorithms: (i) alg07, in scientific paper the authors use some cue words to contextualize the paper; (ii) alg10, the text, section and paragraph generally starts with the main sentence of the text; and (iii) alg14, The title need represent the text as good as possible.

The *f*-measure presents a surprise. Besides alg10 and alg14, the alg16 archive good results. It means that the relationship among sentences in the text are more important than in previous datasets.

Finally, in relation to execution time we find the following conclusions:

- Alg6 is the slower in almost all cases.
- The alg02 and alg13 greatly increased execution time in the last evaluation. This is mainly because they make computations with the words, and texts of the last dataset have more words.
- In general, sentence scoring methods are faster. This is because they use the sentence structure, unlike the word scoring (which makes computations using the words) and graph scoring (which creates a graph structure before running the algorithm).
- The blog dataset has the lower execution time because the texts here have fewer words in relation to the others.
- Alg16 usually is not fast because it has to create the graph and makes computations with words.

5. How can sentence scoring results be improved?

The sentence scoring algorithms are becoming increasingly mature. Consequently, the scientific community is now trying to improve their results rather than creating other algorithms. The six most common issues encountered are Orasan (2009), Nenkova and McKeown (2011): (i) Morphological transformation; (ii) Stop words; (iii) Similar semantics; (iv) Co-reference; (v) Ambiguity, and (vi) Redundancy. The following sections explain each of the strategies listed above and present possible solutions.

5.1. Morphological transformation

Constantin Orasan (2009) points three morphological transformations that improve word scoring methods.

Truncation: It retains only the first six characters of words are kept in an attempt to identify tokens derived from the same root.

Stemming: It is a transformation that builds the basic forms of words, i.e. strips off the plural "s" from nouns, the "ing" from verbs, or other affixes. A stem is a natural group of words with equal (or similar) meaning. After the stemming process, every word is represented by its stem. For instance, the verbs "traveling and "traveled are both transformed into "travel;

Lemmatization: This transformation identifies the lemma of a word. For example, it maps the verbs onto their infinitive and nouns onto their singular form. Thus, the form of the word must be known. Lemmatization requires more resources than the other two methods. It can deal with irregular words by using lists of exceptions.

Reference (Orasan, 2009) presents that the listed transformations improve the summarization results.

5.2. Stop words

The problem addressed here is how to deal with words with little meaning to the text, such as articles, conjunctions, and prepositions. Besides them, words with both high and low frequencies of occurrence are also considered as stop words. There are many tools, such as RetriBlog[6] and Lucene,[7] that provide the removal of stop words. It is important to notice that some stop words could be significant to text summarization, however. For example, some prepositions could refer to important text subjects (co-reference).

Almost all current summarization systems treat stop words (Lloret & Palomar, 2012; Barrera & Verma, 2012; Wei, 2012; Abuobieda et al., 2012) in some way.

5.3. Similar semantics

Words of similar semantics usually mean synonyms. However, relations such as *hypernyms* and *hyponyms* are also important to improve the semantic treatment. Hypernym relationships occur when words are related in some level of a semantic tree. For example, "pet and "dog. A "dog is a type of a "pet, thus they are related. In the problem of sentence scoring, words with similar semantics could be considered as one, increasing the relative importance that word as concept in the text.

There are three main approaches to deal with this problem. The first one is use WordNet[8] relations to verify the similarity between two given words. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. It also includes hypernyms and hyponyms. Reference (Pedersen, Patwardhan, & Michelizzi, 2004) provides an overview of similarity using WordNet. Some examples of summarization systems that use WordNet are (Lloret & Palomar, 2012; Barrera & Verma, 2012; Zhang, Ma, Niu, Gao, & Song, 2012; Gupta et al., 2011).

The second approach that deals with semantic similarity is known as *lexical chains* (Barzilay & Elhadad, 1997). This approach exploits the intuition that topics are expressed using not a single word but different related words, instead. For example, the occurrence of the words such as "car", "wheel", "seat", "passenger" indicates that the text is related to the automobile topic, even if each of the words does not appear very frequently in the text. In other words, this strategy clusters words together and the sentence scoring algorithms

[6] http://sourceforge.net/projects/retriblog/.
[7] http://lucene.apache.org/core/.
[8] http://wordnet.princeton.edu/.

analyze topics or concepts, rather than words in isolation. References (Wei, 2012; Gupta et al., 2011) use this approach.

The last approach is Latent Semantic Analysis (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). This is an unsupervised technique based on word co-occurrences to implicit represent text semantics, that tries to map which words usually appear together (Hachey, Murray, & Reitter, 2006; Balahur et al., 2009).

5.4. Co-reference

Co-reference is the process of matching all references to the same entity in a document, regardless of the syntactic form of the reference. It usually matches noun, full noun phrase or pronoun. Some work have demonstrated that co-reference resolution can be used for substantially improve summarization systems that rely on word frequency features (Nenkova & McKeown, 2011).

A simple example is the use of pronominal reference. For example, *"John will travel tomorrow. He bought the ticket yesterday"*. In this case the pronoun *"he"* refers to *"John"*. Thus, if the words are scored together, they may be more significant. This type of analysis is not widely used in summarization systems because of the performance and accuracy issues.

5.5. Ambiguity

Ambiguity, also known as polysemy, occurs when the same word can have different meanings in different contexts. For example, "apple" could mean a fruit or a computer company. Thus, the sentence score algorithms can assign higher values for some words improperly. Lexical chains may solve this problem.

Two fundamental issues must be taken into account in the context of summarization. Usually in single document summarization, words are in the same context. Here, the probability of ambiguity is low. On the other hand, in the context of multi-document summarization, such a problem may happen, but solving ambiguity may increase performance related problems.

5.6. Redundancy

Unlike the previously presented problems, redundancy is related to sentences and not only to words. Redundancy occurs when multiple sentences have the same content. In general, it is perceived as improper because of its use of duplicative or unnecessary wording, mainly in summaries.

The two techniques that are commonly used to treat this problem are:

Sentence fusion: It is the task of taking two sentences that contain some overlapping information, but that also have fragments that are different, and producing a sentence that conveys the information in common between the two sentences (Krahmer, Marsi, & van Pelt, 2008).

Textual entailment: It consists of determining if the meaning of one text snippet (the hypothesis) can be inferred by another one (the text) (Glickman, 2009). The identification of these entailment relations helps a summarization system avoid incorporating redundancy in final summaries.

These techniques are mainly used into abstractive summarization, but they may be adapted for extractive ones.

6. Conclusions

This paper explains and implements the most important text summarization strategies found in the literature in the last ten years. Three different corpora were used to assess the techniques presented. We selected the five best results obtained with the different test sets, one would obtain a coincidence of four methods as being the best ones: Word Frequency (Alg 1), TF/IDF (Alg 2), Lexical Similarity (Alg 6), and Sentence Length (Alg 9). The strategy "Text-Rank Score" (Alg 16) was also chosen by as providing good results for two of the three data sets tested. The results provided using ROUGE for the quantitative assessment of summarizers was quite close to the ones obtained by the qualitative analysis. The calculus of TF/IDF is by far the most computationally intensive of all methods tested (Alg 2). Methods Word Frequency (Alg 1) and Sentence Length (Alg 9) provide the best balance in execution-time performance and electing relevant sentences. Strategies to compose the results obtained to yield even better summaries are being currently investigated.

Acknowledgements

The research results reported in this paper have been partly funded by a R& D project between Hewlett–Packard do Brazil and UFPE originated from tax exemption (IPI-Law n 8.248, of 1991 and later updates).

References

Ahuobieda, A., Salim, N., Albaham, A. T., Osman, A. H., & Kumar, Y. J. (2012). Text summarization features selection method using pseudo genetic-based model. In *International conference on information retrieval knowledge management* (pp. 193–197).

Baeza-Yates, Ricardo, & Ribeiro-Neto, Berthier (1999). *Modern information retrieval* (1st ed.). Addison Wesley.

Balahur, Alexandra., Lloret, Elena., Boldrini, Ester., Montoyo, Andrés., Palomar, Manuel., & Martínez-Barco, Patricio. (2009). Summarizing threads in blogs using opinion polarity. In *Proceedings of the workshop on events in emerging text types* (pp. 23–31).

Barrera, Araly, & Verma, Rakesh (2012). Combining syntax and semantics for automatic extractive single-document summarization. In *Proceedings of the 13th international conference on computational linguistics and intelligent text processing* (pp. 366–377). Springer-Verlag.

Barzilay, Regina., & Elhadad, Michael. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization* (pp. 10–17).

Deerwester, Scott, Dumais, Susan T., Furnas, George W., Landauer, Thomas K., & Harshman, Richard (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal ACM, 16*(2), 264–285.

Fattah, Mohamed Abdel, & Ren, Fuji (2009). Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech and Language, 23*(1), 126–144.

Glickman, Oren (2009). *Applied textual entailment: A generic framework to capture shallow semantic inference*. VDM Verlag.

Gupta, P., Pendluri, V. S., & Vats, I. (2011). Summarizing text by ranking text units according to shallow linguistic features. In *13th International conference on advanced communication technology* (pp. 1620–1625).

Hachey, Ben., Murray, Gabriel., & Reitter, David. (2006). Dimensionality reduction aids term co-occurrence based multi-document summarization. In *Proceedings of the workshop on task-focused summarization and question answering* (pp. 1–7).

Haque, Rejwanul, Naskar, Sudip Kumar, Way, Andy, Costa-jussa, Marta R., & Banchs, Rafael E. (2010). Sentence similarity-based source context modelling in pbsmt. In *Proceedings of the 2010 International conference on asian language processing* (pp. 257–260). IEEE Computer Society.

Hu, Meishan., Sun, Aixin., & Lim, Ee-Peng. (2007). Comments-oriented blog summarization by sentence extraction. In *Proceedings of the 16th ACM conference on information and knowledge management* (pp. 901–904).

Hu, Meishan, Sun, Aixin, & Lim, Ee-Peng (2008). Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 291–298). New York, NY, USA: ACM.

Krahmer, Emiel., Marsi, Erwin., & van Pelt, Paul. (2008). Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies* (pp. 193–195).

Kulkarni, U. V., & Prasad, Rajesh S. (2010). Implementation and evaluation of evolutionary connectionist approaches to automated text summarization. In *Journal of Computer Science* (pp. 1366–1376). Science Publications.

Lin, Chin-Yew (2004). Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens (Ed.), *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.

Lins, Rafael Dueire., Simske, Steven J., Cabral, Luciano de Souza., Silva, Gabriel de Freitas., Lima, Rinaldo., Mello, Rafael F., & Favaro, Luciano. (2012). A multi-tool scheme for summarizing textual documents. In *Proceedings of 11st IADIS International conference WWW/INTERNET 2012* (pp. 1–8).

Liu, Xiaoyue, Webster, Jonathan J., & Kit, Chunyu (2009). An extractive text summarizer based on significant words. In *Proceedings of the 22nd international conference on computer processing of oriental languages. Language technology for the knowledge-based economy* (pp. 168–178). Berlin, Heidelberg: Springer-Verlag.

Lloret, Elena, & Palomar, Manuel (2009). A gradual combination of features for building automatic summarization systems. In *Proceedings of the 12th International conference on text. Speech and dialogue* (pp. 16–23). Berlin, Heidelberg: Springer-Verlag.

Lloret, Elena, & Palomar, Manuel (2012). Compendium: A text summarisation tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, 1–40 [FirstView].

Lloret, Elena, & Palomar, Manuel (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review*, 37(1), 1–41.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2), 159–165.

Mariño, José B., Banchs, Rafael E., Crego, Josep M., Gispert, Adrià, Lambert, Patrik, Fonollosa, José A. R., et al. (2006). N-gram-based machine translation. *Computational Linguistics*, 32(4), 527–549.

Mihalcea, Rada., & Tarau, Paul. (2004). TextRank: Bringing order into texts. In *Conference on empirical methods in natural language processing, Barcelona, Spain.*

Murdock, Vanessa Graham. (2006). Aspects of sentence retrieval. Ph.D. thesis, University of Massachusetts, Amherst.

Nenkova, Ani, & McKeown, Kathleen (2011). Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3), 103–233.

Nenkova, Ani, & McKeown, Kathleen (2012). A survey of text summarization techniques. In *Mining text data* (pp. 43–76). Springer.

Orasan, Constantin (2009). Comparative evaluation of term-weighting methods for automatic summarization. *Journal of Quantitative Linguistics*, 16, 67–95.

Pedersen, Ted, Patwardhan, Siddharth, & Michelizzi, Jason (2004). Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004* (pp. 38–41). Stroudsburg, PA, USA: Association for Computational Linguistics.

Prasad, Rajesh Shardanand, Uplavikar, Nitish Milind, Wakhare, Sanket Shantilalsa, Jain, Vishal, Yedke & Tejas Avinash (2012). Feature based text summarization. *International Journal of Advances in Computing and Information Researches*, 1.

Satoshi, Chikashi Nobata., Satoshi, Sekine., Murata, Masaki., Uchimoto, Kiyotaka., Utiyama, Masao., & Isahara, Hitoshi. (2001). Keihanna human info-communication. Sentence extraction system assembling multiple evidence. In *Proceedings 2nd NTCIR workshop* (pp. 319–324).

Tonelli, Sara., & Pianta, Emanuele. (2011). Matching documents and summaries using key-concepts. In *Proceedings of the french text mining evaluation workshop.*

Wei, Yang. (2012). Document summarization method based on heterogeneous graph. In *9th International conference on fuzzy systems and knowledge discovery (FSKD)* (pp. 1285–1289).

Zhang, Dongmei., Ma, Jun., Niu, Xiaofei., Gao, Shuai., & Song, Ling. (2012). Multi-document summarization of product reviews. In *9th International conference on fuzzy systems and knowledge discovery (FSKD)* (pp. 1309–1314).

A Platform for Language Independent Summarization

Luciano Cabral[a,b], Rafael Dueire Lins[a], Rafael Mello[a], Fred Freitas[a], Bruno Ávila[a], Steven Simske[c], and Marcelo Riss[d]

[a] Federal University of Pernambuco, Recife, Brazil

[b] Federal Institute of Pernambuco, Caruaru, Brazil

[c] Hewlett-Packard Labs., Fort Collins, CO 80528, USA

[d] Hewlett-Packard Brazil, Porto Alegre, Brazil

{lsc4,rdl,rflm,fred,bta}@cin.ufpe.br, {steven.simske,marcelo.riss}@hp.com

ABSTRACT

The text data available on the Internet is not only huge in volume, but also in diversity of subject, quality and idiom. Such factors make it infeasible to efficiently scavenge useful information from it. Automatic text summarization is a possible solution for efficiently addressing such a problem, because it aims to sieve the relevant information in documents by creating shorter versions of the text. However, most of the techniques and tools available for automatic text summarization are designed only for the English language, which is a severe restriction. There are multilingual platforms that support, at most, 2 languages. This paper proposes a language independent summarization platform that provides corpus acquisition, language classification, translation and text summarization for 25 different languages.

Categories and Subject Descriptors

I.2.7 **[Artificial Intelligence]**: Natural Language Processing – *Language understanding, Machine translation and Text analysis*.

General Terms

Measurement, Experimentation and Algorithms.

Keywords

Multilingual Summarization, Classification, Translation, Platform.

1. INTRODUCTION

The amount of data available on the web is growing every year. There are millions textual documents published in different languages making of the web an open environment for content dissemination. It is increasingly difficult to efficiently find useful information in the Internet. Thus, it is necessary to use language independent methods to understand, classify and present, clearly and concisely the existing information in different languages, saving users resources and time. Text summarization has been pointed out as a possible solution to the document classification problem and creates a shorter version of a document with its essential content. Most automatic document summarization techniques and tools were developed for the English language. This paper presents a platform for language independent summarization that combines techniques for language identification, content translation and summarization. The proposed solution first pre-processes the input text to make it treatable by the following modules. Then, language identification is performed followed by translation into English and then summarization or direct summarization. The results are then analyzed to yield the final summary.

DocEng'14, September 16–19, 2014, Fort Collins, Colorado, USA.

Other works found in the technical literature aim to perform multilingual summarization. MEAD, by Radev and his colleagues [1], makes use of only 8 summarization algorithms and was assessed only with the Chinese and English languages. Evans and his collaborators [2] use sentence similarity and clustering, as a summarization strategy and was focused on Arabic and English. Roark and Fisher [3] use Machine Learning to obtain a query-focused sentence ranking (with supervision). In reference [3] they describe an experiment with translated documents which bears some resemblance to the strategy proposed here. However, their work neither explicitly mentions the number of supported languages nor brings the idea of having more than one translation of the same document as a way to compensate the semantic loss of the translation process as introduced here. Litvak, Last and Friedman [4] more recently, use genetic algorithms in the summarization task. Similarly to the aforementioned multi-language summarization algorithms, their work supports only two languages (English and Hebrew). Gupta [5] uses a hybrid algorithm for summarization, supporting Hindi and Punjabi docs.

Unfortunately, none of the multilingual summarization references listed above offer elements for testing their performance in a common data set or corpus making difficult an independent assessment or performance analysis. Despite this, they focus on a disjoint set of languages (Arabic, Chinese, Hebrew, Hindi and Punjabi), while the focus here is on the languages spoken by the European Union. As they address only specific languages: no language identification module is part of their solutions.

2. PLIS - ARCHITECTURE DESCRIPTION

This section presents the functionality of the main modules of the Platform for Language Independent Summarization. Figure 1 sketches the PLIS architecture.

The initial task of the platform is text pre-processing in which the input text has all its non-textual parts removed and sentences numbered and organized one sentence per line. Then, the resulting text undergoes the language identification phase. If the text is in English it is submitted to the extractive summarization module, which selects the most significant sentences from the original text after pre-processing using the several sentence scoring methods described in the literature acknowledged as the most efficient ones for extractive summarization. Otherwise, the text is submitted to

some language independent summarization algorithms and to several tools that will translate each of the sentences in the original text into English. As the automatic translation process is likely to introduce semantic losses to the original text the use of more than one translation tool may compensate such losses. The several versions of the translated text are submitted to the extractive summarization module yielding for each input a set of numbers, each of them related to the sentences in the original text. The different sets of sentences chosen are analyzed by the Sentence Scoring and Selection Module, which will produce a new set of indices that correspond to the summary, encompassing the chosen sentences in the original text.

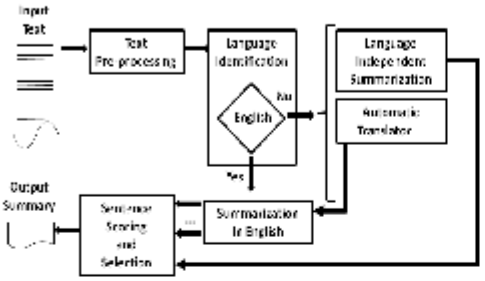

Figure 1. The PLIS Architecture.

The following sections detail the functionality of the main modules of the PLIS architecture.

2.1 Language classification

This module implements the CALIM language identification algorithm, described in reference [6], which works with all the 25 languages in use in the European Community today. Such algorithm was inspired on the ideas of Dunning [7], Cavnar and Trenkle [8] and Lins and Gonçalves [9], based on language profiles containing the most frequent words (≈250 per language). Beyond your respective relative frequency in a total of 25 languages (English, Portuguese, Spanish, Italian, French, German, Bulgarian, Czech, Danish, Dutch, Estonian, Finnish, Greek, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, Slovenian, Swedish, Arabic, Hebrew, Hindi and Korean.). To help in the profile creation task were used some language databases provided by Lexiteria [10], which is an initiative aiming at understanding various aspects of human language.

Besides that, due to performance reasons, only words with maximum length ≤ 5 were considered. This choice is justified because, in most languages, the most frequent words like prepositions, personal pronouns, etc. are also the shorter ones. Thus, in case a word is longer than 5 characters, was taken its 5-length suffix to be included in the language profile. Other values were considered for average length in our experiments, but the best performance results were achieved with $length \leq 5$ [11]. The Language Identification Module of PLIS applies a heuristic that contributed to more accurate results in the experiments performed. This heuristic assumes that if a word (or token) comprises very specific n-grams exclusively found in certain language (such as "*do*" in Portuguese), then the method assigns a greater value than the normal vote, which is equal to 1. The language with the highest accumulative scoring value will be chosen as the best matching language for the document.

At the end, each token from the input text will be classified in one or more languages. In case of two or more languages ended in a draw, proceeds to with an additional scoring step consisting in multiplying the final score of each token by its confidence rating. It is calculated by the ratio between the token frequency and the sum of all token frequencies for each profile, so this simple heuristic contributed to choose the right language.

The Europarl v7 "full" corpus [12] with about 60,000 documents with a random distribution between documents and the 21 official languages from the European Community was used to test the accuracy of the Language Identification Module and provided an accuracy of 99.992 %. The other four languages implemented (Arabic, Hebrew, Hindi and Korean) use alphabet-based identification yielding 100% recognition accuracy.

2.2 Automatic Translation Module

This task performs an intermediate automatic translation process using Microsoft API [13]. Such function is a fundamental contribution, which aggregates to the platform the condition of supporting some languages, making it language independent. Simple to use, this API is free for requests up to 2,500 characters, which generated an initial difficulty, solved by the textual fragmentation before the translation process to proceed with the defragmentation after processing.

Frame 1. Translation sample. (a) Original.

[1] (CNNMéxico) — El jamaicano Usain Bolt, que consiguió este domingo su tercera medalla de oro en Moscú, su octava medalla de oro en campeonatos del mundo y la décima en total, se dijo orgulloso de sí mismo y anunció que seguirá trabajando "para dominar tanto tiempo como sea posible".

[2] "Da gusto vencer", dijo, luego de ganar con el equipo jamaicano el primer lugar en la carrera de relevos 4x100, según EFE.

...

[14] En días anteriores, durante las actividades del Mundial de Atletismo en Moscú, Rusia, Bolt recuperó su corona en los 100 metros y también se posicionó como el mejor en los 200.

[15] La última victoria de Bolt, al igual que la de sus compatriotas, llega como aire fresco para el deporte en Jamaica, sacudido en los últimos meses por escándalos de dopaje como el del velocista Asafa Powell.

(b) Performed by Microsoft Translation API.

[1] (CNNMéxico) — the Jamaican Usain Bolt, who won on Sunday their third gold medal in Moscow, its eighth gold medal in the World Championships and tenth overall, said proud of itself and announced that it will continue to work "to dominate as long as possible".

[2] "It gives taste to beat," he said, after winning first place with the Jamaican team in the 4 x 100, according to EFE relay race.

...

[14] In earlier days, during the activities of the Athletics World Cup in Moscow, Russia, Bolt regained his Crown in the 100 meters and also ranked as the best in the 200.

[15] The last victory of Bolt, as well as of their compatriots, arrives as fresh air for the sport in Jamaica, shaken in recent months by scandals of doping as the sprinter Asafa Powell.

In fact, despite translation differences, the sentence indices are maintained, which helps in the multi-language summarization strategy, which concerns in: translating the original content to English; to run the summarization process; and finally, with the results, perform a mapping between the sentences of the obtained summary and the original document. It provides at the final, an abstract in original language of the document.

As sentences are individually numbered by the Text Pre-Processing Module and the summarization process chooses the number of a sentence, instead of the sentence itself, the platform will work correctly even if the language does not follow the left-to-right and top-to-bottom way of writing which is the standard for European languages. In Hebrew and Arabic, for instance, two languages that are also in the PLIS architecture, texts are written right-to-left and top-to-bottom.

2.3 Summarization

This step aims to automatically create a small version of the original document. It tries to extract the meaning of the text. This work uses only combined extractive methods combined. The choice of the methods to be implemented was directed by the work in reference [14], which evaluates some methods in different corpora. The methods presented better results and processing time were chosen to compose this task, they are:

- *Word Frequency:* The most frequently occurring words in a text have the highest score. In other words, sentences containing the most frequent words in a document stand a higher chance of being selected for the final summary. The assumption is that the higher the frequency of a word in the text, the more likely that it indicates the subject of the text.
- *Sentence Length*: This feature is employed to penalize sentences that are either too short or long. These sentences are not considered an optimal selection. The method considers length as the number of words in a sentence.
- *Sentence Position*: Many approaches use sentence position as a score criterion. According to Abuobieda *et al.* [15], the first sentence in the paragraph is considered an important sentence and a strong candidate to be included in the summary; Grupta [16] says that the first sentences of paragraphs and words in titles and headings are more relevant to summarization. The method proposed in [17] assigns score 1 to the first N sentences and 0 to the others, where N is a given threshold for the number of sentences.

In the processing, each approach compute values for the sentences of the text, these values are aggregated and ranked; the most punctuated sentences are selected for the summary according the threshold provided by user, which may be the sentence quantity (e.g. 6) or percentage of the text size (i.e. if the original contains 20 sentences, 30% corresponds to 6 sentences).

3. EXPERIMENTAL RESULTS

This section presents the methodology experimental results and its analysis for assessing the quality of the summaries generated by the PLIS. The experiments used three different corpora, which have different languages: CNN-English, CNN-Spanish and TeMário-Portuguese. The corpora have the following characteristics: news containing only the text, i.e. neither figures nor videos; high quality news text written by professionals; and a gold-standard summary generated by humans.

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [18] was used for assessing the quality of the summaries generated by the platform. It is a quantitative method based on *n*-gram statistics and it is highly correlated with human evaluations [19]. This fully automated evaluator essentially measures the content similarity between system-generated summaries and the corresponding gold-standard summaries (generated by humans). The evaluation is performed using the *n*-gram (1,1) setting of ROUGE, because it was found to have the highest correlation with human judgments at a confidence level of 95%.

In the following sections, the experimental results and its analyses are presented separately for each corpus.

3.1 CNN-English

The CNN-English corpus developed by Lins and his collaborators [20] contains news articles extracted from the CNN website. The current version of this corpus presents 400 texts assigned to 10 categories: Asia, business, Europe, Latin America, Middle East, US, sports, technology, travel and world news.

The experimental results of ROUGE for the CNN-English dataset are shown in Table 1. For comparison purposes, some of the experimental results obtained by Ferreira and his colleagues reported in reference [14] were used here. They assessed 17 different summarization algorithms, including those used in this work, using the same CNN-English corpus. They labeled their implementations of the word frequency, length and position of the sentence as Alg01, Alg09 and Alg10, respectively.

Table 1. ROUGE results for CNN-English dataset.

Summarizer	Avg Recall	Avg Precision	Avg F-measure
PLIS	**0.71 (±0.24)**	0.29 (±0.13)	0.41 (±0.16)
Alg01 [14]	0.71 (±0.19)	0.35 (±0.13)	0.46 (±0.15)
Alg09 [14]	0.70 (±0.18)	0.33 (±0.12)	0.44 (±0.15)
Alg10 [14]	0.61 (±0.22)	**0.40 (±0.13)**	**0.47 (±0.15)**

As one may observe in Table 1 the current strategy in the Summarization Module of PLIS achieved the best recall result, similarly to the word frequency scoring strategy (Alg 01) in [14]. Alg10 (sentence position) has the best precision and F-measure. PLIS showed slightly lower values in precision and F-measure than the other summarization algorithms analyzed. Although the platform loses on average precision, it maintains the average recall for the best result (Alg01). The analysis of better tuning strategies of the sentence scoring algorithms to compose results in the Summarization Module of PLIS are under study.

3.2 CNN-Spanish

The CNN-Spanish corpus developed in this work followed the same development path of the CNN-English corpus [20]. It contains news articles extracted from the CNN Mexico website. The current version of this corpus presents 400 texts assigned to 08 categories: sports, entertainment, world, national, opinion, technology, travel and health news.

The experimental results of ROUGE for CNN-Spanish dataset are shown in Table 3. Since this corpus is introduced here, then there no experimental results from other summarizers to analyze.

Table 2. ROUGE results for CNN-Spanish dataset.

Summarizer	Avg Recall	Avg Precision	Avg F-measure
PLIS	**0.72 (±0.11)**	0.11 (±0.03)	0.20 (±0.05)

The average recall of PLIS is close to the values achieved by other corpora presented in Sections 3.1 and 3.3. It might indicate that the amount of errors introduced by the intermediate translation step did not affect the overall process. Nevertheless, further experiments are being developed to yield more accurate figures.

3.3 TeMário-Portuguese

TeMário test collection [21] contains 100 news articles from the Brazilian newspapers: *Jornal do Brasil* and *Folha de São Paulo*. The documents were selected to cover a variety of domains (e.g. world, politics, foreign affairs, editorials) and an expert in Brazilian Portuguese language produced manual summaries.

The experimental results of ROUGE for the TeMário dataset are shown in Table 2. For comparison purposes, some of the experimental results obtained by Leite and Rino [22] were used here. They used the same TeMário-Portuguese corpus in their experiments aiming to assess the combination of multiple machine learning features for automatic summarization. They used the measure ROUGE-1 with 30% compression rate and the manual summaries were used as gold. However, only the average recall was reported. For a fair comparison, the experiments used the same configuration.

Table 3. ROUGE results for TeMário-Portuguese dataset.

Summarizer	Avg Recall	Summarizer	Avg Recall
PLIS	**0.77**	TextRank [22]	0.51
SuPor2-LR [22]	0.53	BestCN [22]	0.50
SuPor-2 [22]	0.52	Baseline [22]	0.49

The average recall of PLIS is the highest one. This means that translating the text to English and then extracting a summary appears to be a *valid* technique for multilingual summarization.

4. CONCLUSIONS

This paper presented a language independent summarization platform that aims to create short versions of documents to help in multilingual content analysis on the web. An architecture using integrated services based on language classification and translation, summarization, where each method was chosen by prior studies according the best results obtained over years. Three different corpora and languages were used to assess the platform.

The main contributions of this paper are: (a) providing an extensible platform to language independent summarization; (b) supporting up to 25 different languages with the translation intermediate process and a summarization-combined method; (c) the evaluation shows compatible results compared other recent works of the similar purpose. In addition, the summarization platform is easily extensible and with differential adding new languages or new summarization methods. Studies to improve the platform are being intensified, aiming to encompass other languages and summarization methods to yield even better results, including in future studies regarding the sensitivity of summarization with respect to translation process.

The strategies used in the Sentence Scoring and Selection Module are in a better tuning process, but depend on having a much larger test corpus. Efforts in such direction are being performed by the authors.

5. ACKNOWLEDGMENTS

The research results reported in this paper have been partly funded by a R&D project between Hewlett-Packard-Brazil and UFPE originated from tax exemption (IPI-Law n 8.248, of 1991 and later updates).

6. REFERENCES

[1] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel e Z. Zhu, "MEAD - a platform for multidocument multilingual text summarization.," *Proceedings of LREC 2004*, Lisbon, Portugal, 2004.

[2] D. K. Evans, K. Mckeown e J. L. Klavans, "Similarity-based Multilingual Multi-Document Summarization," *IEEE Transactions on Information Theory*, vol. 49, 2005.

[3] B. Roark e S. Fisher, "OGI/OHSU baseline multilingual multi-document summarization system," *IEEE International Conference on Microelectronic Systems Education*, , USA, 2005.

[4] M. Litvak, M. Last e M. Friedman, "A New Approach to Improving Multilingual Summarization Using a Genetic Algorithm," *48th Annual Meeting of the Assoc for C. Linguistics*, Sweden, 2010.

[5] V. Gupta, "Hybrid Algorithm for Multilingual Summarization of Hindi and Punjabi Documents," *Lecture Notes in Computer Science. Mining Intelligence and Knowledge Exploration*, vol. 8284, pp. 717-727, 2013.

[6] L. Cabral, R. Lins, R. Lima, R. Ferreira, F. Freitas, G. Silva, G. Cavalcanti, S. Simske e L. Favaro, "A Hybrid Algorithm for Automatic Language Detection on Web and Text Documents," *11th IAPR International Workshop on Document Analysis Systems. Tours – Loire Valley, France*, 10 April 2014.

[7] T. Dunning, "Statistical identification of language," Technical Report CRL MCCS-94-273, Computer Research Lab, New Mexico University, New Mexico, 1994.

[8] W. B. Cavnar e J. M. Trenkle, "N-Gram Based Text Categorization.," *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161-169, 1994.

[9] R. Lins e P. Gonçalves, "Automatic language identification of written texts," em *Proceedings of the ACM Symposium on Applied Computing (SAC '04)*, New York, NY, USA, 2004.

[10] Lexitaria, "Word Frequency Lists," Lexitaria, 2002. [Online]. Available: http://www.lexitaria.com/. [Acesso em 09 10 2013].

[11] L. Cabral, R. Lins, R. Lima e S. Simske, "A comparative assessment of language identification approaches in textual documents," *IADIS International Conference Applied Computing 2012*, Madrid, 2012.

[12] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," *MT Summit 2005*, 2005.

[13] Microsoft Corporation, "Microsoft Translator V2," MSDN, 2014. Available: http://msdn.microsoft.com/en-us/library/ff512423.aspx. [Last acess 10 March 2014].

[14] R. Ferreira, L. Cabral, R. Lins, G. Silva, F. Freitas, G. Cavalcanti, R. Lima, S. Simske e L. Favaro, "Assessing sentence scoring techniques for extrative text summarization," *Expert Systems with Applications*, pp. 5755-5764, 2013.

[15] A. Abuobieda, N. Salim, A. Albaham, A. Osman e Y. Kumar, "Text summarization features selection method using pseudo genetic-based model.," *International Conference on Information Retrieval Knowledge Management (CAMP)*, pp. 193-197, March 2012.

[16] P. Gupta, V. Pendluri e I. Vats, "Summarizing text by ranking text units according to shallow linguistic features," *13th International Conference on Advanced Communication Technology (ICACT)*, pp. 1620-1625, February 2011.

[17] C. N. Satoshi, S. Satoshi, M. Murata, K. Uchimoto, M. Utiyama, H. Isahara e K. Human, "Info-communication: Sentence extraction system assembling multiple evidence.," *Proceedings of 2nd NTCIR Workshop*, pp. 319-324, 2001.

[18] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries.," em *Text summarization branches out: Proceedings of the ACL-04 workshop*, Barcelona, Spain, Stan Szpakowicz Marie-Francine Moens, 2004, pp. 74-81.

[19] C.-Y. Lin e E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," *Proc. of Human Language Technology Conference (HLT-NAACL 2003)*, Canada, 2003.

[20] R. D. Lins, S. J. Simske, L. S. Cabral, G. F. P. Silva, R. J. Lima, R. F. Mello e L. Favaro, "A multi-tool scheme for summarizing textual documents," em *Proceedings of 11st IADIS International Conference WWW/INTERNET*, Madrid, Spain, 2012.

[21] T. Pardo e L. Rino, "TeMario: a corpus for automatic text summarization.," Technical report, NILC-TR-03-09., São Paulo, 2003.

[22] D. Leite e L. Rino, "Combining multiple features for automatic text summarization through Machine Learning," em *Computational Processing of the Portuguese Language: 8th International Conference, PROPOR 2008*, Springer-Verlag , 2008, pp. 122-132.