Identificador Automático de Acrônimo Sem Explicação

Henrique Papa[1], **Márcio de Souza Dias**[1]

[1]Departamento de Computação - Universidade Federal de Goiás (UFG)
Regional Catalão. Catalão/GO.

henrique_ahg@hotmail.com, marciosouzadias@ufg.br

Abstract. *Automatically generated multidocument summaries can present a number of language quality issues. These problems compromise the reader's understanding of the content. Therefore, this work proposes the development of a prototype of an automatic identifier for one of the errors of linguistic quality, called Acronym No Explanation. Using a multidocument summary corpus, the prototype obtained an accuracy of 98.7 % in error identification.*

Resumo. *Os sumários multidocumento gerados automaticamente podem apresentar diversos problemas relacionados à qualidade linguística. Esses problemas comprometem a compreensão do conteúdo por parte do leitor. Diante disso, este trabalho propõe o desenvolvimento de um protótipo de um identificador automático para um dos erros de qualidade linguística, chamado Acrônimo Sem Explicação. Utilizando um córpus de sumários multidocumento, o protótipo obteve uma acurácia de 98,7% na identificação do erro.*

1. Introdução

Atualmente, a escrita é uma das principais formas de comunicação entre as pessoas, sendo utilizada em meios como jornais, livros, artigos, redes sociais, etc. Todos os dias, uma enorme quantidade de conteúdo escrito é produzida, principalmente na Internet. Devido a isso, a Sumarização Automática Multidocumento (SAM) vem ganhando destaque na comunidade científica. O processo de SAM consiste em otimizar a obtenção das principais informações contidas em diversos textos, sobre o mesmo assunto, em apenas um [Mani 2001].

Apesar dos avanços alcançados, os sumarizadores automáticos ainda não tratam de forma satisfatória os aspectos linguísticos que afetam a coesão e a coerência textual, prejudicando, por consequência, a compreensão do conteúdo por parte do leitor [Nenkova et al. 2011]. Devido a isso, alguns estudos como [Koch 1998], [Otterbacher et al. 2002], [Friedrich et al. 2014] e [Dias 2016], focam na identificação de erros linguísticos em textos e em sumários gerados automaticamente. Tal identificação vem sendo feita de forma manual, tornando esta tarefa bastante onerosa. Por isso, o desenvolvimento de técnicas capazes de identificar de forma automática tais erros se fez necessário para um futuro pós-processamento dos geradores e sumarizadores automáticos.

Dias (2016) utilizou sumários automáticos multidocumento do Português do Brasil, presentes no córpus CSTNews [Cardoso et al. 2011], para identificar 12 erros linguísticos. Dentre eles, o erro de Acrônimo Sem Explicação foi o erro de maior frequência. Este erro ocorre quando um acrônimo é mencionado em um texto sem que

haja uma explicação referente ao seu significado no próprio texto. Desta forma, a compressão do leitor pode ser prejudicada, uma vez que este pode não conhecer o seu significado. O exemplo na Figura 1 ilustra que o acrônimo "ONU" (em negrito) não foi devidamente explicado no sumário.

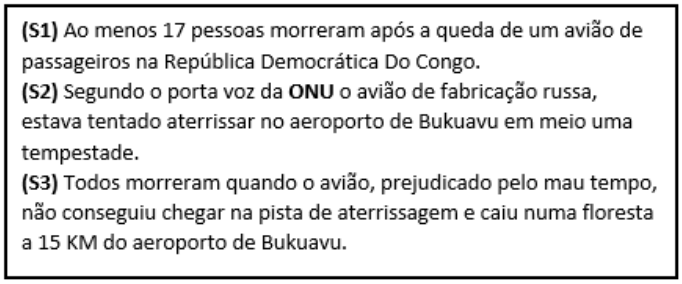
(S1) Ao menos 17 pessoas morreram após a queda de um avião de passageiros na República Democrática Do Congo.
(S2) Segundo o porta voz da **ONU** o avião de fabricação russa, estava tentado aterrissar no aeroporto de Bukuavu em meio uma tempestade.
(S3) Todos morreram quando o avião, prejudicado pelo mau tempo, não conseguiu chegar na pista de aterrissagem e caiu numa floresta a 15 KM do aeroporto de Bukuavu.

Figura 1. Sumário multidocumento com Acrônomo sem Explicação [Dias 2016].

Dado o problema do Acrônimo Sem Explicação, o qual foi anotado manualmente [Dias 2016], não encontramos na literatura trabalhos que o identifique de forma automática. Assim, nós propomos neste trabalho identificar automaticamente todos as ocorrências de acrônimos que não tiveram sua explicação dada em um sumário multidocumento.

Este artigo está organizado da seguinte maneira: na Seção 2, há uma breve descrição sobre os trabalhos relacionados; a Seção 3 apresenta o córpus utilizado; na Seção 4, a metodologia de desenvolvimento é apresentada; na Seção 5, os experimentos e resultados alcançados são discutidos; na Seção 6, por fim, uma breve conclusão é apresentada.

2. Trabalhos Relacionados

Este trabalho se baseou na necessidade de melhorar a qualidade linguística dos sumários gerados automaticamente. Para isso, é importante, inicialmente, identificar os erros que prejudicam tal qualidade no processo de sumarização.

Dentre os trabalhos que procuraram identificar erros que afetam a qualidade linguística, o trabalho de [Koch 1998] relata a importância de conectores (elementos gramaticais, lexicais, sintáticos) de coesão na manutencão da qualidade linguística. Sabe-se que o bom uso desses conectores linguísticos entre as sentenças favorece a compreensão e a interpretação do texto como um todo.

O trabalho desenvolvido por [Otterbacher et al. 2002] concluiu que os principais problemas linguísticos encontrados nos sumários automáticos estão relacionados à falta de pontuação, uso de sentenças muito longas e uso inadequado de parênteses ou outros elementos textuais. [Friedrich et al. 2014], por sua vez, apresentaram um córpus de sumários multidocumento, chamado LQVSumm. Nesse estudo, dois erros foram identificados e tratados, o primeiro é referente à menção de entidades(relacionado a problemas de referência) e o outro envolve erros de gramática e redundância.

Diante de tais problemas, [Dias 2016] desenvolveu um modelo para classificar a coerência textual em sumários multidocumento para o Português do Brasil utilizando aprendizado de máquina. Além disso, o autor fez um levantamento de erros linguísticos

que afetam diretamente a coerência dos sumários multidocumento gerados automaticamente.

3. Córpus

Neste trabalho, o córpus[1] utilizado foi o CSTNews. Tal córpus contém 50 coletâneas de textos jornalístico extraídos de jornais importantes no país ("O Globo", "Jornal do Brasil", "Gazeta do Povo", etc). Os textos extraídos desses jornais versam sobre os seguintes temas: Mundo, política, cotidiano, ciência e esporte. Cada coletânea possui de 2 a 3 textos de diferentes origens. Ao todo são 140 textos contendo em média 334 palavras.

O CSTNews é um córpus rico de informações anotadas por especialistas da área linguística e linguística computacional. Dentre as várias anotações presentes no córpus, o CSTNews possui uma anotação de erros linguísticos nos sumários multidocumento automáticos, oriundos dos textos fonte do CSTNews [Dias 2016]. Para essa tarefa de anotação foram utilizados 200 sumários gerados automaticamente por 4 sumarizadores (GistSumm [Filho et al. 2007], RSumm [Ribaldo 2013], RC-4 [Cardoso et al. 2015] e MTRST-MCAD [Castro Jorge 2015]) e vários pesquisadores entre linguistas e cientistas de computação. Para cada uma das 50 coleções do córpus, cada sumarizador gerou um sumário. A Tabela 1 mostra os dados do córpus de sumários automáticos.

Tabela 1. Dados do córpus [Dias 2016]

Sumarizador	Média de palavras	Média de senteças
GistSumm	362	11
RSumm	134	4
RC-4	132	4
MTRST-MCAD	139.78	7.92

4. Metodologia de Desenvolvimento

O processo de desenvolvimento do protótipo consistiu em 3 etapas: o pré-processamento do córpus, a identificação de acrônimos e a verificação de explicação. Nas subseções a seguir, cada etapa é explicada.

4.1. Pré-processamento

O primeiro passo do pré-processamento foi remover todas as anotações de erros que os sumários multidocumento possuem, uma vez que as marcações dos erros foram feitas em um processo de anotação realizada por pesquisadores da área de computação e da linguística [Dias 2016](ver Figura 2). Tal procedimento é necessário, uma vez que, o objetivo desse trabalho é identificar automaticamente o erro de Acrônimo Sem Explicação.

Em seguida, os sumários foram submetidos a um processo de segmentação sentencial e a um processo de tokenização. Assim, cada sentença foi segmentada em uma linha do arquivo e realizou-se a separação de palavras e caracteres especiais em uma lista para cada sumário. Na Figura 3 é ilustrado um trecho de um sumário antes e depois da etapa de pré-processamento.

[1]Coleção de material escrito e/ou falado usado no estudo da língua. `https://dictionary.cambridge.org/pt/dicionario/ingles/corpus?q=crpus`

[S1] Ao menos 17 pessoas morreram após a queda de um avião de passageiros na República Democrática do Congo.

[S2] Segundo uma porta-voz da **<e TYPE=ACR-EXP SC="Organização das Nações Unidas">**ONU**</e>**, o avião, de fabricação russa, estava tentando aterrissar no aeroporto de Bukavu em meio a uma tempestade.

Figura 2. Córpus anotado [Dias 2016]

Antes	Depois
[S1] As operações da Companhia Paulista de Trens Metropolitanos (CPTM) seguiam normais. [S2] Não ha infomações sobre o estado de saúde dos envolvidos.	Lista_S1[“As”, “operações”, “da”, “Companhia”, “Paulista”, “de”, “Trens”, “Metropolitanos”, “(“, “CPTM “, “)”, “seguiam” ,“normais” , “.”] Lista_S2[“Não”, “há”, “informações”, “sobre”, “o”, “estado”, “de” ,“saúde”, “dos”, “envolvidos”, “.”]

Figura 3. Resultado do pré-processamento

O resultado do pré-processamento é uma lista de *tokens* para cada sentença do sumário.

4.2. Identificação de Acrônimos

A identificação de acrônimos é a parte essencial no processo de análise automática. Nesta fase, o protótipo percorre cada elemento da lista de *tokens* criada na etapa de pré-processamento, aplicando dois métodos para identificar acrônimos: o método simples e o composto.

No método simples, identificamos como acrônimo palavras que estão em caixa alta e sem acentos. No método composto, por sua vez, utilizamos pesquisas na web para verificar se determinado termo pode ser considerado um acrônimo ou não. No entanto, sabe-se que ambos os métodos são suscetíveis a erros, uma vez que alguns *tokens* são erroneamente identificados como acrônimos. Esses *tokens*, geralmente, se referem a cidades, símbolos monetários e abreviações. Em virtude disso, elaboramos algumas regras que descartam os tipos de *tokens* acima mencionados. Em seguida, há uma breve descrição das regras e métodos utilizados.

Regra das Cidades

Em um texto, os nomes das cidades podem estar em caixa alta e, dessa forma, serem identificados erroneamente como acrônimos. Para solucionar esse problema, a “regra das cidades” analisa os primeiros elementos da lista de *tokens* que estão escritos em caixa alta. O processo se repete até que se encontre outro *tokens* que não esteja em caixa alta. Caso a condição de parada seja o símbolo “-“, a regra impede que o *token* seja considerado um acrônimo. Na Figura 4 é mostrado a regra das cidades.

Observando a figura 4, nota-se que os tokens “SÃO” e “PAULO” seguidos de um

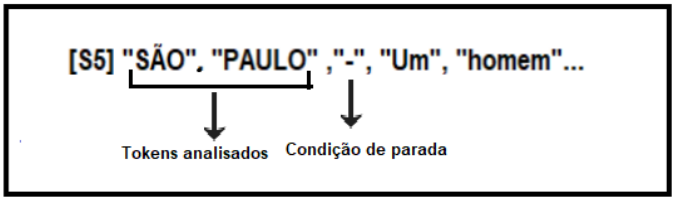

Figura 4. Regra das cidades

hífen caracterizam a citação de uma cidade no início da sentença. Dessa forma, apresentam a sintaxe que se encaixa na regra das cidades.

Regra das Moedas

Símbolos monetários como o US$ (representação do dólar) poderiam ser considerados acrônimos pelo fato de todas as letras estarem maiúsculas. Assim, essa regra é para evitar esse tipo de situação. A regra da moeda verifica se a próxima posição da lista é o símbolo "$", caso seja, a regra impede que o *token* seja considerado acrônimo.

Regra das Reduções

Redução é uma maneira simplificada de escrever uma palavra, por exemplo: televisão é reduzida ou abreviada como TV. Tratar esses casos foi um dos maiores desafios enfrentados durante a implementação deste protótipo, pois diferente das demais regras, não encontramos uma sintaxe específica no texto quanto ao uso de abreviações.

Para este cenário, utilizamos uma função que realiza o acesso e busca automaticamente na página da Academia Brasileira de Letras [2] afim de encontrar as principais reduções da língua portuguesa. Entretanto, apenas tal procedimento não é suficiente em casos ambíguos, ou seja, um *token* pode ser um acrônimo ou uma abreviação, dependendo do contexto. Por exemplo, o *token* FMM (pode representar Fundo da Marinha Mercante ou força magnetomotriz).

Para reduzir as chances de gerar ambiguidades, apenas os *tokens* compostos por duas letras foram considerados pela regra das reduções, com exceções dos acrônimos de estados brasileiros como SP (São Paulo) e de partidos políticos como PT (Partido dos Trabalhadores). A informação referente aos acrônimos de partidos políticos foi obtida por meio da página do TSE (Tribunal Superior Eleitoral)[3]. Ao fim de todo esse processo, uma lista de abreviações foi criada. Desta forma, compara-se cada *token* a lista de abreviações para verificar se tal *token* pode considerado acrônimo ou não.

Método simples

Conforme mencionado, as regras funcionam como filtros e são aplicadas para eliminar *tokens* que podem ser detectados como acrônimos quando na verdade não são. Uma vez que um *token* passa por todas as regras, ele será analisado pelos métodos simples e composto, os quais determinam de fato se o elemento é realmente um acrônimo.

O método simples averigua apenas a sintaxe dos *tokens*. Dessa forma, esse método verifica se um dado elemento é composto apenas por letras, se possui no mínimo duas

[2]http://www.academia.org.br/nossa-lingua/reducoes
[3]http://www.tse.jus.br/partidos/partidos-politicos/registrados-no-tse

letras e se todas as letras estão em maiúsculas e sem acentos. Feito isso, o *token* que atender a todos esses requisitos será considerado um acrônimo.

Método Composto

O método simples consegue identificar de forma correta grande parte dos acrônimos, porém alguns acrônimos como Infraero (Empresa Brasileira de Infraestrutura Aeroportuária) nem sempre são escritos em letras maiúsculas. Devido a isso, o método composto foi desenvolvido para identificar siglas que estão fora dos padrões considerados nas regras e no método simples.

O método composto faz uso de uma ferramenta que permite realizar pesquisas na Wikipédia [4].Este processo gera um alto custo computacional, por isso ele é utilizado apenas em último caso quando as regras e o método simples não conseguem dar uma resposta correta sobre a identificação de um acrônimo.

O processo de pesquisa é realizado por meio dos seguintes passos: Inicialmente, o método analisa se a primeira letra do *token* é maiúscula.Em caso positivo, o *token* é utilizado na pesquisa da Wikipédia (utilizou-se a API - Wikipédia1.4.0). Em seguida, um texto contendo o item pesquisado é retornado e a primeira sentença do texto encontrado é extraída. Por fim, o método analisa se o *token* encontra-se entre parênteses ou no padrão "***TOKEN*** (<*significado do token*>)"na sentença extraída do texto da Wikipédia. Dessa forma, o *token* será considerado acrônimo. Na Figura 5 é mostrado o passo a passo de todo este processo ao pesquisar por Anac.

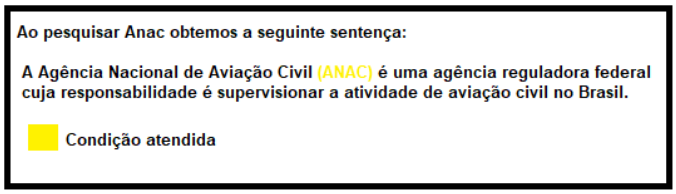

Figura 5. Método Composto

4.3. Verificar Explicação

A última etapa do processo de identificação do erro Acrônimo Sem Explicação consiste em verificar se um acrônimo foi explicado ou não corretamente. Dessa forma, o processo de verificação busca o significado do acrônimo no próprio texto, a partir de duas formas: antes ou após a menção do acrônimo.

Em ambos os casos, para auxiliar o processo de verificação, criou-se uma lista contendo as letras de cada acrônimo, pois a explicação deve conter uma palavra com a inicial de cada letra da lista. Além disso, foi estabelecido, de forma empírica, um limite de *tokens* para a explicação. Esse limite é determinado pelo número de letras mais 5 elementos (conectivos que ligam as explicações de cada acrônimo). Esse valor de 5 deve-se a possibilidade da explicação conter *tokens* como: "de", "e", "da", "sigla" e "para".

O processo de verificação antes averigua se o significado de um acrônimo foi citado anteriormente a sua menção. Para isso, inicialmente, a função procura um indicativo

[4]https://pt.wikipedia.org

de explicação, o caractere "(". Uma vez encontrado esse indicativo, cria-se uma lista de letras formada pelos caracteres que compõem o acrônimo na ordem inversa de escrita. Por exemplo, o acrônimo "ONU" gera a seguinte lista de letras: "U","N","O". Em seguida, percorre-se os *tokens* anteriores ao indicativo "(", comparando-se o caractere inicial de cada *tokens* com o primeiro campo da lista de letras. A cada palavra encontrada a primeira posição da lista de letras é removida. Assim, ao término da lista de letras a função determina que a sigla foi explicada corretamente.

A verificação depois é um processo semelhante. As únicas diferenças são: o sentido da verificação, que ocorre após a menção da sigla e a forma como é feita a lista de letras, a qual segue o sentido de escrita. Por exemplo, o acrônimo "ONU" terá a seguinte lista de letras: "O","N","U". A Figura6 a seguir demonstra ambos os tipos de verificação.

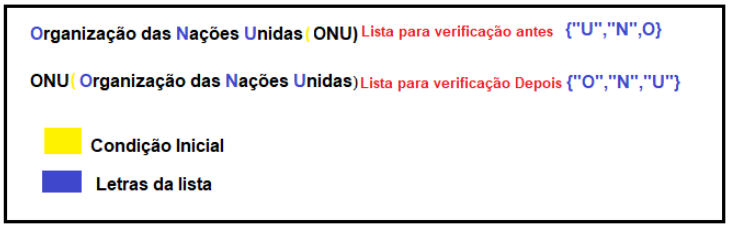

Figura 6. Verificando explicação

5. Experimentos e Resultados

Para realizar os experimentos, os sumários automáticos multidocumento do córpus CST-News escolhidos tinham que possuir o erro de Acrônimo Sem Explicação. Desta forma, 92 sumários dos 200 foram utilizados. Com o intuito de elaborar, verificar a utilidade e medir a eficiência das regras e métodos desenvolvidos, os 92 sumários foram divididos em treino e teste, sendo que o córpus de treino ficou com 68 sumários (74% dos sumários considerados) e o córpus de teste com 24 sumários (26% dos sumários considerados).

O córpus de teste contém 77 anotações referentes ao erro acrônimo sem explicação, destas o protótipo construído detectou corretamente 76 erros. Desta forma, os resultados da fase de teste foram considerados satisfatórios, tendo em vista que o protótipo atingiu 98,7% de acurácia, cometendo apenas um erro ao não identificar o acrônimo Cofins (Contribuição para o Financiamento da Seguridade Social).

A falha ocorreu no método de verificação composto. Tendo em vista que o acrônimo Cofins não é muito comum. Devido a isso a busca atribui o significado a uma cidade de Minas Gerais não a um acrônimo

6. Conclusão

Este trabalho inovou ao implementar um protótipo que permite identificar automaticamente o erro linguístico, Acrônimo Sem Explicação, considerado um dos erros mais frequentes na sumarização automática multidocumento. Com a acurácia de 98,7%, acreditamos que as heurísticas desenvolvidas mostraram-se eficientes e diversificadas, mesmo com uma quantidade sumários não tanto expressiva. Para trabalhos futuros seria interessante aplicar o protótipo em textos maiores e de outros gêneros. Além disso, adaptar o

protótipo na identificação de outros tipos de erros linguísticos, assim como aprimorar a eficácia desse protótipo para ser utilizado em uma aplicação web.

Referências

Cardoso, P., Castro Jorge, M., and Pardo, T. (2015). Exploring the rhetorical structure theory for multi-document summarization. In *Proceedings of the 5th Workshop RST and Discourse Studies*, pages 1 – 10.

Cardoso, P., Mazieiro, E., Jorge, M., Seno, E., di Felippo, A., Rino, L., Nunes, M., and Pardo, T. (2011). Cstnews - a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105.

Castro Jorge, M. L. R. (2015). *Modelagem gerativa para sumarização automática multidocumento*. PhD thesis, Instituto de Ciências Matemáticas e de Computação - ICMC/USP.

Dias, M. S. (2016). *Investigação de modelos de coerência local para sumários multidocumento*. PhD thesis, Instituto de Ciências Matemática e de Computação - Universidade de São Paulo.

Filho, P. P. B., Pardo, T. A. S., and das Graças Volpe Nunes, M. (2007). Sumarização automática de textos científicos: Estudo de caso com o sistema gistsumm. Technical report, NILC - ICMC-USP. 23 p.

Friedrich, A., Valeeva, M., and Palmer, A. (2014). Lqvsumm: A corpus of linguistic quality violations in multi-document summarization. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Koch, I. G. V. (1998). *A coesão textual – Mecanismos de Constituição Textual, A organização do Texto, Fenômenos de Linguagem*. Linguística Contexto – Repensando a Língua Portuguesa, 10 edition.

Mani, I. (2001). *Automatic summarization*, volume 3. John Benjamins Publishing.

Nenkova, A., McKeown, K., et al. (2011). Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Otterbacher, J. C., Radev, D. R., and Luo, A. (2002). Revisions that improve cohesion in multi-document summaries: A preliminary study. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, pages 27–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ribaldo, R. (2013). Investigação de mapas de relacionamento para sumarização multidocumento. Monografia de Conclusão de Curso, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. São Carlos-SP, Novembro, 61p.