Assessing the Impact of Stemming Algorithms Applied to Brazilian Legislative Documents Retrieval

Ellen Souza[1,2], Gyovana Moriyama[2], Douglas Vitório[1,3], André C. P. L. F. de Carvalho[2], Nádia Félix[2,4], Hidelberg O. Albuquerque[1,3], Adriano L. I. Oliveira[3]

[1]MiningBR Research Group, Federal Rural University of Pernambuco
CEP: 52171-900 – Recife/PE – Brazil

[2]Institute of Mathematics and Computer Sciences, University of São Paulo, Brazil

[3]Centro de Informática, Federal University of Pernambuco, Brazil

[4]Institute of Informatics, Federal University of Goiás, Brazil

ellen.ramos@ufrpe.br, gymori@usp.br, andre@icmc.usp.br

{damsv,hoa,alio}@cin.ufpe.br, nadia.felix@ufg.br

Abstract. *The main purpose of stemming is to reduce the inflected words into its root form or stem. Thus, words can be mapped to the same concept, improving the process of information retrieval, regarding its ability to index documents and to reduce data dimensionality. However, the efficiency of those algorithms varies according to different aspects. Also, studies in the field area reached contrasting conclusions. This work assesses the use of stemmers in the retrieval of legislative documents written in Portuguese. Four stemmers together with BM25 were evaluated in two legislative corpora from the Brazilian Chamber of Deputies. RSLP-S and Savoy stemmers showed the best improvements in the information retrieval pipeline.*

1. Introduction

Information retrieval (IR) looks for unstructured material from within large collections, satisfying an information need [Manning et al. 2008]. Law was one of the first knowledge areas to adopt IR, with the first domain-specific legal retrieval system appearing as early as 1960 [Maxwell and Schafer 2008]. The importance of legal applications created a sub-area of IR, *Legal IR*, which covers a large variety of legal texts, including legislation, case law, and scholarly works [Maxwell and Schafer 2008].

In the last years, due to the huge amount of information available, which continues to increase rapidly, improved IR techniques have become necessary [Moral et al. 2014]. Thus, stemming algorithms, which can generate concise word representations, has been largely used in information retrieval systems [Alvares et al. 2005]. When used for IR, stemming can improve predictive accuracy and reduce computational costs, being one of the first steps in the IR pipeline [Moral et al. 2014, de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018].

A stemming algorithm, also called stemmer, extracts the morphological root, stem, of a word. For such, a stemmer removes affixes that carry grammatical or lexical information about the word [Moral et al. 2014]. A stemmer can: (i) cluster words according

to their topic, as many words are derivations from the same stem and can be considered as belonging to the same concept; (ii) index the documents in an IR process, according to their topics, as their terms are grouped by stems (that are similar to concepts), and (iii) reduce the collection of documents to a set of topics or stems, which can both reduce the space needed to store the structures used by an IR system and the computational load [Moral et al. 2014].

The efficiency of stemmers varies according to the language used with and the application domain [Alvares et al. 2005]. Studies evaluating the effects of stemming for IR reached contrasting conclusions [Orengo and Huyck 2001]. Researchers compared well-known stemming algorithms for texts in the English language and did not find any significant improvement due to the use of stemming, with an increase in recall and reduction in precision [Orengo and Huyck 2001]. However, authors agreed on its benefits in specific contexts, such as when the language is highly inflective (the case of the Portuguese language), when documents are short or when there is limited space for storing data [Alvares et al. 2005]. Some researchers also argue that the nature of the documents can influence its predictive performance [Alvares et al. 2005].

This work investigates the effect of the use of stemmers in the retrieval of legislative documents written in the Brazilian Portuguese language. It also investigates how the predictive performance in an IR system is affected by using dimensionality reduction techniques. To the best of the authors knowledge, this is the first evaluation of Portuguese stemming algorithms using texts from the legislative domain. The reported research is part of the *Ulysses* project, an institutional set of artificial intelligence initiatives to increase transparency, improving the Brazilian Chamber of Deputies relationship with the Brazilian population, and supporting the legislative activity [Almeida 2021].

This paper is organized as follows: Sec. 2 presents the major related studies. Sec. 3 details the IR pipeline used in this study. Sec. 4 presents the experiments performed and discusses the obtained results. Sec. 5 brings the conclusion and points out future works.

2. Related Work

We found few papers investigating the application of stemming for IR, specifically for the Portuguese language. In [Orengo et al. 2006], the authors evaluated three stemming algorithms for texts in Portuguese: Porter, RSLP, and RSLP-S. According to the experimental results, RSLP-S was the best algorithm in terms of MAP (Mean Average Precision) and Pr@10 (Precision at 10 documents). Experiments were carried out with CLEF 2006's dataset, which contains texts from Público and Folha de São Paulo newspapers.

In [Flores et al. 2010, Flores and Moreira 2016], the authors evaluated the benefits of stemming in four different languages, one of them Portuguese. They compared 8 stemmers, 5 specifically designed for the Portuguese language (Porter, RSLP, RSLP-S, Savoy, and StemBR) and 3 language-independent ones (Linguistica, GRAS, and Stemmer-S). The algorithms were evaluated in two different ways: 1) by their quality, using the Paice's method, which uses the metrics of Overstemming Index (OI), Understemming Index (UI), Stemming Weight (SW), and Error Rate Relative to Truncation (ERRT); and 2) by their impact on document retrieval effectiveness. For the Portuguese language, in the first evaluation, the best algorithms, using ERRT, were RSLP and Porter. In the second evaluation, using MAP, Savoy presented the best predictive performance. All the experiments used

datasets from the CLEF's tracks of 2005 and 2006, whose Portuguese corpus contains articles from the Brazilian newspaper Folha de São Paulo.

Using jurisprudential data from the Supreme Court of the State of Sergipe, the experiments reported in [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018] also used Porter, RSLP, RSLP-S, and Savoy for two stemming evaluations. They considered the average number of unique terms obtained by each stemmer and its average percentage of reduction, in which the RSLP proved to be the best. The stemmer's impact on the legal document retrieval was evaluated in terms of MAP, MPC (average of Pr@10), and MRP (average of R-Precision) and using the BM25 IR algorithm. The best algorithms for this task were RSLP-S and Savoy, as they reduced the dimensionality of the data and increased the effectiveness of Information Retrieval. However, the authors pointed out that the use of radicalization usually deteriorated the Okapi BM25 performance.

Although this last study focuses on the legal domain, the texts used are from judgments and monocratic decisions of Appeals Court, and judgments and monocratic decisions of Special Courts, which differ in size, entities, and vocabulary from the ones we are using from the Brazilian Chamber of Deputies (see Sec. 3.1).

3. Method

3.1. Corpora

The Brazilian House of Representatives processes approximately 30 thousand bills[1] every year. Each bill needs to be formalized as an initial legislative document *draft* and an optional justification document. For a typical bill, a large number of documents, in different formats, is produced and submitted to the Legislative Consulting (CONLE), an advisory body of the House, whose main role is to provide the support to the law making process.

The process starts through *job requests* (legislative consultations). The *job requests* are the queries and represent the user's input to the system. While the bills and other *job requests* are the output answer, ranked according to a matching rate between the documents and the query. Thus, two legislative corpora were used to build and validate this research: the *Bills* and the *Job Request* corpora. The former is made available, while the latter has confidential information and cannot be made available[2]. Both are detailed in the following subsections.

3.1.1. Bills

For the experiments, the three most common types of bills were selected: Law Project (Projeto de Lei - PL), Complementary Law Project (Projeto de Lei Complementar - PLC), and Constitutional Amendment Proposal (Proposta de Emenda Constituicional - PEC). As a result, the final corpus had 48,555 proposals. The attribute *imgArquivoTeorPDF*, which is the bill itself, was used in the experiments. It has an average of 300 words.

[1] Legislative Information System - SiLeg

[2] https://drive.camara.leg.br/s/c3p2nLgLRcMz6eX

3.1.2. Job Request

This corpus represents the user query and contains 295 anonymized *Job Requests* from 2019. Data identifying the parliamentarian who made the *Job Request* to CONLE were removed. This corpus has two attributes: *NUMERO-PROPOSICAO-SILEG* and *TxTAssunto*. The former contains the number of the SiLeg bill that was originated from the *Job Request* specified in the latter attribute. Table 1 shows examples of parliamentarians' job requests. Most job requests have between 10 and 40 words and other files may be attached to it, such as: images, spreadsheets, links, and other documents.

Table 1. Samples from anonymized Job Request corpus.

NUMERO-PROPOSICAO-SILEG	TxTAssunto
PL XXXX/2019	Projeto para restabelecer na CLT a proibição de terceirização para atividade fim (Project to prohibit the outsourcing of core activity in the CLT)
PL XXXX/2019	Criação de PL, com base nos dois esboços encaminhados anexo. (Make of bill based on the two sketches sent in the attachment)
PL XXXX/2019	Solicito parecer pela aprovação de acordo com a solicitação XXXX/AAAA. (Request an opinion for the approval according to job request number XXXX/AAAA.)
PL XXXX/2019	Complementar parecer em função da apensação do PL XXXX/AA ao mesmo (Complementary opinion according to the PL XXXX/AA)
PL XXXX/2019	Parlamentar solicita aprovação (Parliamentarian requests approval)

3.2. Preprocessing

Both corpora presented in previous subsections had their texts converted to lowercase and the punctuation was removed. After this, every word was represented by its stem [Hotho et al. 2005], according to the evaluated algorithm. All preprocessing steps were performed using the Python NLTK library. The pipeline is available here[3]. Table 2 shows the application of four stemming algorithms used in the experiments.

- NoStem: generates no reduction of terms.
- Porter: originally written for the English language, in 1980, and adapted to the Portuguese language later. Porter is a full stemming algorithm that is based on a series of 5 conditional rules that are applied in sequence to remove the suffixes [Porter 1980]. We used the Snowball implementation available on NLTK.
- RSLP (Removedor de Sufixos da Lingua Portuguesa): a rule-based algorithm developed by [Orengo and Huyck 2001] and later improved in 2006 [Orengo et al. 2006]. Like Porter, it also applies successive steps to remove the suffixes. However, as it was developed specially for the Portuguese language, it has more rules than Porter. It has 8 steps and also presents a list of exception which prevents the algorithm from removing suffixes of words that have endings that are similar to suffixes. It was called STEMP, before.
- RSLP-S: is a variation of the RSLP algorithm, which applies only the first rule of RSLP that deals with the plural reduction. We implemented the algorithm in the Python language, based on [Orengo et al. 2006].
- Savoy (UniNE): developed by Jacques Savoy in 2006, this algorithm presents stemmers for various languages, including Portuguese. It is simpler than the others, as it has less rules. It removes inflections attached to both nouns and adjectives, based on rules for the plural and feminine form. We implemented the algorithm in the Python language, based on [Savoy 2006].

[3]https://github.com/Convenio-Camara-dos-Deputados/BM25-Experiments

Table 2. Example of stemming for each algorithm used in the experiments

NoStem	projeto	estado	solicito	deputado	aprovação	criação	federal
Porter	projet	estad	solicit	deput	aprov	criaçã	federal
RSLP (STEMP)	projet	est	solicit	deput	aprov	cri	feder
RSLP-S	projeto	estado	solicito	deputado	aprovação	criação	federal
Savoy (UniNE)	projet	estad	solicit	deputad	aprovaca	criaca	federal

3.3. Information Retrieval

Best Match 25 (BM25) [Robertson et al. 1994] is the most well-known scoring function for "bag of words" document retrieval [Kamphuis et al. 2020]. It is derived from the binary independence relevance model to include within-document term frequency information and document length normalization in the probabilistic framework for IR [Robertson and Zaragoza 2009]. The algorithm has also been used successfully in the retrieval of legal documents [N de Oliveira and C Junior 2018, Gomes and Ladeira 2020, Chalkidis et al. 2021]. We have implemented two variants presented in [Trotman et al. 2014] using the Python language.

The Okapi BM25's [Robertson et al. 1994] scoring function estimates the relevance of a document d to a query q, based on the query terms appearing in d, regardless of their proximity within d: where q_i is the ***i-th*** query term, with $idf(q_i)$ inverse document frequency and $tf(q_i, d)$ term frequency. BM25L [Lv and Zhai 2011] is built on the observation that the Okapi variant penalizes more longer documents compared to shorter ones. It *shifts* the term frequency normalization formula to boost scores of very long documents.

3.4. Evaluation

3.4.1. Stemming algorithms evaluation

As in [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018], to assess the algorithm's capacity of dimensionality reduction, we considered the average number of unique terms (UDT) obtained by each stemmer and its average percentage of reduction (RP). Those measures are computed as:

- Unique Terms (UDT_s) = Frequency of unique terms after document stemming.
- Average of unique terms: $\mu = (UTD_{S1} + UTD_{S2} + +UTD_{Sn})/n$.
- Reduction percentage: $RP_R = 100 - (UTD_S \times 100)/UTD_{NoStem}$.
- Average of reduction percentage: $\mu = (RP_{S1} + RP_{S2} + ... + RP_{Sn})/n$.

3.4.2. Information retrieval evaluation

In our corpora, we have only a list of relevant documents. Therefore, we have evaluated the results in terms of *Recall* (R), which is the fraction of relevant documents that are retrieved. We analyzed the results from R@1 to R@20 (Recall at 1 document to Recall at 20 documents).

4. Results and discussion

The next subsections present and discuss the results for the stemming algorithms evaluation (Section 4.1) and for the analysis of their impact in the IR task (Section 4.2).

4.1. Stemming algorithms evaluation

Figure 1 presents the average number of unique terms per document (UTD) obtained per stemmer in the Bills corpus (1A) and in the Job Request corpus (1B). The results in both corpora were the same, in which the RSLP algorithm showed the largest reduction of unique terms: it decreased the dimensionality almost by 50, in average, when dealing with bills, and by 2.5 dealing with job requests. RSLP-S, in its turn, presented the smallest dimensionality reduction, while Porter and Savoy achieved similar UTD. This finding was the same as that one obtained by [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018], when dealing with jurisprudential data. In their work, RSLP also showed the best capacity in terms of UTD, while RSLP-S was the worst.

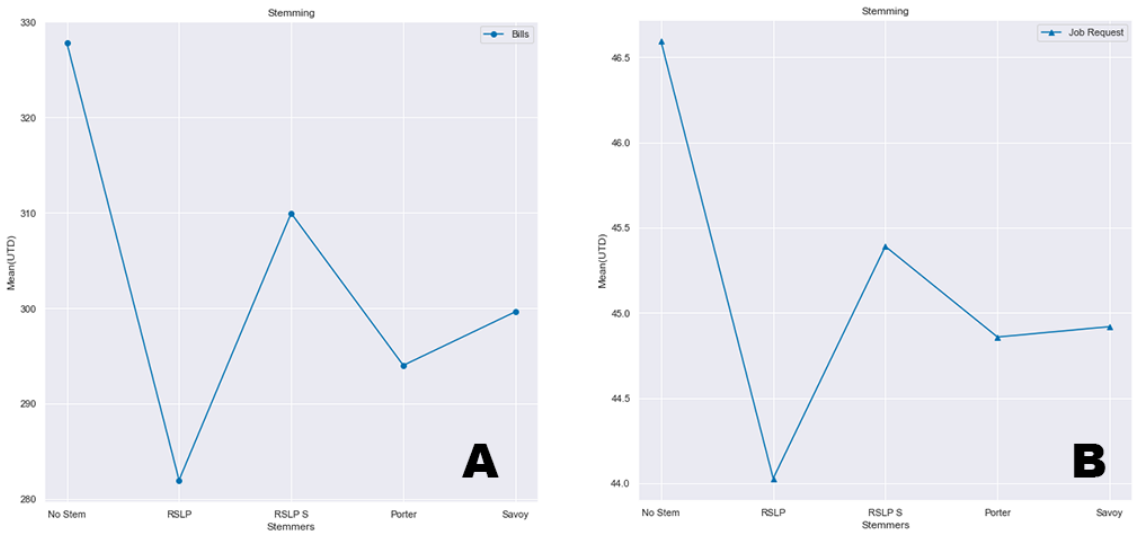

Figure 1. Average UTD per document obtained by each stemmer in the Bills corpus (A) and the Job Request corpus (B).

By the analysis of the average percentage of reduction per document (RP) in both datasets (Figure 2) we could also confirm the findings of [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018]. The RSLP algorithm achieved the best percentage of reduction as well.

The studies of [Orengo et al. 2006] and [Flores et al. 2010, Flores and Moreira 2016] did not analyze the stemming algorithms using the same metrics as we did (UTD and RP); however, in their experiments, RSLP was considered the best stemmer in terms of reduction of terms and Error Rate Relative to Truncation.

So, we can conclude that RSLP is the most effective stemmer in terms of dimensionality reduction for the legislative corpora analyzed, while RSLP-S presented the worst results. These results confirm the ones found in the literature using datasets from other domains, including jurisprudence. In addiction, it is worth mentioning that the RSLP-S performance may be explained due to how it works: by focusing only on plural reduction.

4.2. Information retrieval evaluation

Assessing the impact of stemming in the IR task, Figure 3 brings the Recall obtained by Okapi BM25 (3A) and BM25L (3B) using each stemmer for the Bills corpus.

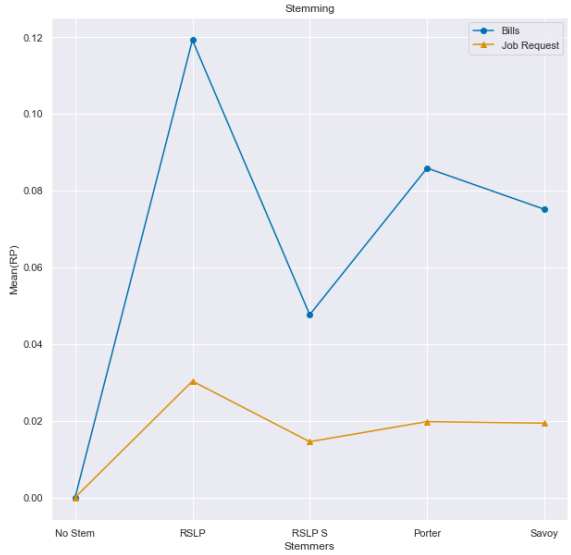

Figure 2. Average RP per document obtained by each stemmer.

According to the Recall@10 results, we could observe that using Okapi BM25 without radicalization (NoStem) achieved better results than using stemming algorithms. This confirms the finding of [N de Oliveira and C Junior 2018], that the use of stemmers deteriorate the original BM25 performance. However, the same was not observed using BM25L variant, for which stemming improved the results. For this IR algorithm, Savoy was the best stemmer in terms of Recall@10. Nevertheless, considering the Recall@20 measure, the Savoy algorithm outperformed the use of IR without radicalization, using the Okapi BM25. While for the use of BM25L, Savoy and RSLP-S were the best algorithms.

We also performed a statistical analysis using the Friedman test [Friedman 1937] and the Nemenyi post-hoc test [Nemenyi 1963], considering the values of Recall@1 to Recall@20 for each stemming algorithm. The Friedman test pointed out that there was a difference between the algorithms for both Okapi BM25 and BM25L, while the Nemenyi post-hoc test indicated which algorithms showed a difference. As we can see in the CD diagrams from Figure 4, for Okapi BM25, IR without stemming (NoStem) achieved the best results, being statistically similar only to Savoy and statistically better than the others. However, for BM25L, the use of Savoy was the best and statistically better than NoStem.

In this sense, we could notice that the adoption of radicalization for legislative documents retrieval depends on the IR algorithm chosen: using Okapi BM25, it is recommended that no stemmer is used; while using BM25L, the dimensionality reduction may improve the IR performance. Meanwhile, analyzing just the performance of the different stemmers, Savoy and RSLP-S can be pointed out as the best ones in terms of Recall for the scenarios analyzed; while RSLP and Porter (Snowball) achieved very poor results.

Finally, comparing these findings with the analysis from Section 4.1, we can state that a great dimensionality reduction does not indicate a better performance for IR. The RSLP algorithm showed the best results in terms of reduction, but when it was used for IR, it deteriorated the BM25 performances. On the other hand, RSLP-S was the worst in terms of UTD and RP, while one of the best for documents retrieval.

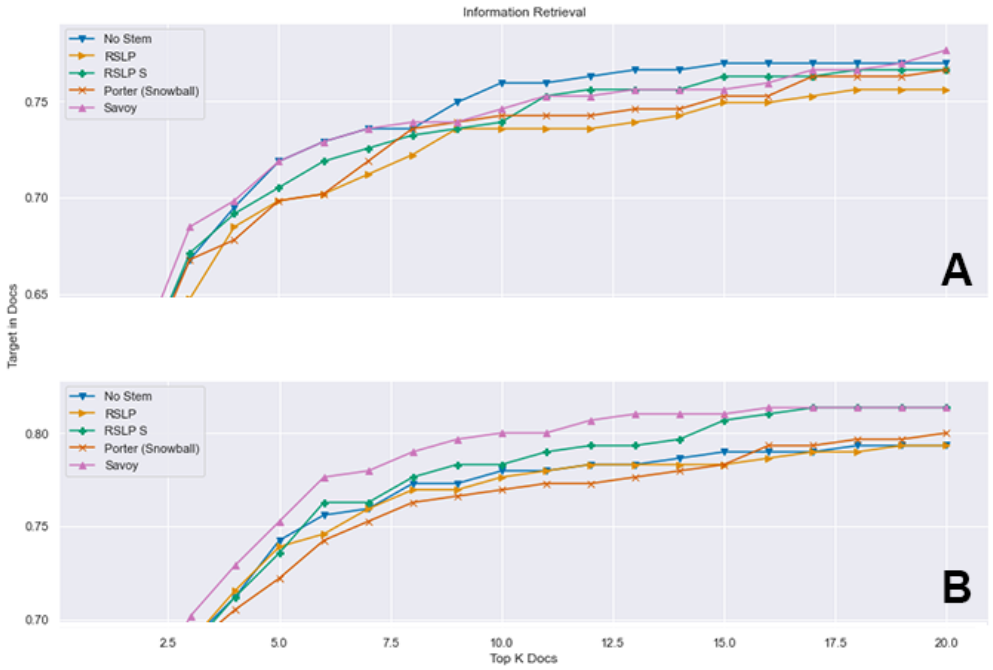

Figure 3. Recall achieved by Okapi BM25 (A) and BM25L (B) using each stemmer.

Figure 4. Results of Nemenyi post-hoc test for Okapi BM25 (A) and BM25L (B).

5. Conclusion

This paper presents a contribution related to the application of stemming algorithms on the legislative domain for terms of data dimensionality reduction and evaluates the efficiency of algorithms in the IR task. Four Portuguese stemmers were evaluated: Porter, RSLP (STEMP), RSLP-S, and Savoy (UniNE). The average number of unique terms per document and the average percentage of reduction per document were used to evaluate the stemmers. For the IR task, two BM25 variants were evaluated using the Recall measure.

The RSLP algorithm showed the largest reduction of unique terms (UDT), while RSLP-S presented the smallest dimensionality reduction. The RSLP also achieved the best percentage of reduction (RP). Assessing the impact of stemming in the IR task with the BM25L, Savoy was the best stemmer in terms of Recall@10, while, using Recall@20, Savoy and RSLP-S achieved the same result. For the Okapi BM25, with Recall@10, we could observe that IR without radicalization (NoStem) achieved better results than using stemming algorithms, confirming the finding of [N de Oliveira and C Junior 2018].

We conclude that the adoption of radicalization for legislative documents retrieval depends on the IR algorithm chosen and that a great dimensionality reduction does not indicate a better performance for IR. For future work, we intend to analyze the impact of the reduction using other corpora in the Legislative domain, measuring its impact on IR.

References

Almeida, P. G. R. (2021). Uma jornada para um Parlamento inteligente: Câmara dos Deputados do Brasil. *Red Información*, 24.

Alvares, R. V., Garcia, A. C. B., and Ferraz, I. (2005). Stembr: A stemming algorithm for the brazilian portuguese language. In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence*, pages 693–701, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chalkidis, I., Fergadiotis, M., Manginas, N., Katakalou, E., and Malakasiotis, P. (2021). Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK legislation where text similarity has limitations. *arXiv preprint arXiv:2101.10726*.

de Oliveira, R. A. and Colaço Júnior, M. (2017). Assessing the impact of stemming algorithms applied to judicial jurisprudence-an experimental analysis. In *International Conference on Enterprise Information Systems*, volume 2, pages 99–105. SCITEPRESS.

Flores, F. N. and Moreira, V. P. (2016). Assessing the impact of stemming accuracy on information retrieval–a multilingual perspective. *Information Processing & Management*, 52(5):840–854.

Flores, F. N., Moreira, V. P., and Heuser, C. A. (2010). Assessing the impact of stemming accuracy on information retrieval. In *International Conference on Computational Processing of the Portuguese Language*, pages 11–20. Springer.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Gomes, T. and Ladeira, M. (2020). A new conceptual framework for enhancing legal information retrieval at the brazilian superior court of justice. In *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, page 26–29.

Hotho, A., Nürnberger, A., and Paaß, G. (2005). A Brief Survey of Text Mining. *Journal for Computational Linguistics and Language Technology*, pages 1–37.

Kamphuis, C., de Vries, A. P., Boytsov, L., and Lin, J. (2020). Which BM25 do you mean? A large-scale reproducibility study of scoring variants. In *Advances in Information Retrieval*, pages 28–34.

Lv, Y. and Zhai, C. (2011). When documents are very long, BM25 fails! In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1103–1104.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, USA.

Maxwell, K. T. and Schafer, B. (2008). Concept and context in legal information retrieval. *Frontiers in Artificial Intelligence and Applications*, 189:63–72.

Moral, C., de Antonio, A., Imbert, R., and Ramírez, J. (2014). A Survey of Stemming Algorithms in Information Retrieval. *Information Research: An International Electronic Journal*, 19(1)(n1):22.

N de Oliveira, R. A. and C Junior, M. (2018). Experimental analysis of stemming on jurisprudential documents retrieval. *Information*, 9(2):28.

Nemenyi, P. (1963). *Distribution-free multiple comparisons*. PhD thesis, Princeton University.

Orengo, V. M., Buriol, L. S., and Coelho, A. R. (2006). A study on the use of stemming for monolingual ad-hoc portuguese information retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 91–98. Springer.

Orengo, V. M. and Huyck, C. R. (2001). A stemming algorithmm for the portuguese language. In *SPIRE*, volume 8, pages 186–193.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 40:211–218.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at TREC-3. In *TREC*.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Savoy, J. (2006). Light stemming approaches for the french, portuguese, german and hungarian languages. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1031–1035, New York, NY, USA. Association for Computing Machinery.

Trotman, A., Puurula, A., and Burgess, B. (2014). Improvements to BM25 and language models examined. *ACM International Conference Proceeding Series*, 27-28-Nove:58–65.