Information Processing and Management 52 (2016) 840–854

Contents lists available at ScienceDirect

Information Processing and Management

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

Assessing the impact of Stemming Accuracy on Information Retrieval – A multilingual perspective

Felipe N. Flores[*], Viviane P. Moreira

Instituto de Informática – UFRGS, Av. Bento Gonçalves, 9500, 91501-970 Porto Alegre, Brazil

ARTICLE INFO

Article history:
Received 31 March 2015
Revised 14 March 2016
Accepted 24 March 2016
Available online 18 April 2016

Keywords:
Stemming
Information Retrieval
Evaluation
Multilingual

ABSTRACT

The quality of stemming algorithms is typically measured in two different ways: (*i*) how accurately they map the variant forms of a word to the same stem; or (*ii*) how much improvement they bring to Information Retrieval systems. In this article, we evaluate various stemming algorithms, in four languages, in terms of accuracy and in terms of their aid to Information Retrieval. The aim is to assess whether the most accurate stemmers are also the ones that bring the biggest gain in Information Retrieval. Experiments in English, French, Portuguese, and Spanish show that this is not always the case, as stemmers with higher error rates yield better retrieval quality. As a byproduct, we also identified the most accurate stemmers and the best for Information Retrieval purposes.

1. Introduction

Stemming is the conflation of the variant forms of a word into a single representation, *i.e.*, the stem. For example, the terms *presentation, presenting*, and *presented* could all be stemmed to *present*. The stem does not have to be a valid word, but it needs to capture the meaning of the words.

Stemming is usually carried out by algorithms that strip word suffixes (but some also strip prefixes) which is why this technique is called affix stripping. Other stemming techniques include the use of dictionaries – which contain the correct form of stemming for the maximum number of words – and statistical stemming. Affix stripping stemmers are language dependent, that is, the rules are designed based on some knowledge of the language. One cannot use stemming rules designed for Portuguese, for example, and expect them to perform well on a French corpus. Statistical stemmers, however, aim at learning the stemming rules automatically and thus eliminating the need of knowing the language (Majumder et al., 2007; Paik, Mitra, Parui, & Järvelin, 2011).

The quality of stemming algorithms is typically assessed in one of these manners: (*i*) how correctly the stemmer maps semantically and morphologically related words to the same stem; or (*ii*) how much improvement the stemmer brings to Information Retrieval (IR). According to Jones and Galliers (1996) and Mollá and Hutchinson (2003), the first would be an *intrinsic evaluation* as it analyzes the accuracy of the results of the stemmer as a stand-alone system. The latter would be an *extrinsic evaluation*, because it analyses the impact of the stemmer in one of its applications.

Stemming is widely used in IR with the aim of increasing recall (*i.e.*, the number of relevant documents retrieved in response to a user query) (Baeza-Yates & Ribeiro-Neto, 2011). Another benefit is the reduction of the size of the index files, because a stem can represent many different words, resulting in fewer distinct index entries. Stemming is also used in other

[*] Corresponding author. Tel.: +55 5191186168.
E-mail address: fnflores@inf.ufrgs.br (F.N. Flores).

http://dx.doi.org/10.1016/j.ipm.2016.03.004
0306-4573/© 2016 Elsevier Ltd. All rights reserved.

contexts, such as data mining, sentiment analysis, text categorization, automatic indexing, text summarization, information extraction, lexical analysis, and several natural language processing tasks. A number of studies report on the effectiveness of using stemming in an IR system, especially for the English language (Harman, 1991; Hull, 1996; Krovetz, 1993).

Since the goal of stemming is to increase recall, in practice, it tends to reduce precision as a side effect – an undesirable feature for Web search. Levene (2010) states that since stemming may not improve the top retrieved documents (*i.e.*, the ones that would appear on the first page of a search engine's result), it is not often used on the Web and large corpora. Still, the author mentions that major search engines such as Google and Yahoo! do employ some form of (light) stemming. Alternatively, rather than doing affix removal during indexing and querying, stemming can be done as a type of query expansion. Peng, Ahmed, Li, and Lu (2007) propose expanding the query with variant forms of the words and deal with stemming on a case by case basis, applying it selectively. The authors report gains in efficiency and quality of the retrieved results.

The interest in investigating stemming algorithms started back in 1960s and is still ongoing. While the first endeavors were devoted to create rule-based suffix strippers, nowadays the focus is on statistical stemmers, which demand no linguistic knowledge of the language for which they are designed. Also, as pointed out by Sharma (2012), once the suffixes to be removed are obtained, stemming can be done faster than applying rule-based stemmers.

Comparing stemming algorithms under different criteria is still performed in recent research. Jivani (2011) summarizes the advantages and disadvantages of known stemmers for English; while Moral, Antonio, Imbert, and Ramirez (2014) performed a more detailed analysis that also includes stemmers for other languages. Sirsat, Chavan, and Mahale (2013) evaluates the strength and accuracy of the most widely used English stemmers. Méndez-Cruz, Torres-Moreno, Medina-Urrea, and Sierra (2013) performed an extrinsic evaluation of stemmers on summarization tasks. The work by Brychcín and Konopík (2015) contemplates both intrinsic and extrinsic metrics in their experimental evaluation. The intrinsic evaluation relied on corpora annotated with the lemmas of the word forms and the extrinsic evaluations were on a standard IR setting. However, the goal was not to compare the two types of evaluation.

Although related, stemming and lemmatization are different tasks. While the former reduces words to their stems, the latter reduces them to their canonical forms (*i.e.,* dictionary form), or *lemmas*. For example, the word *having* would be lemmatized to *have* and stemmed to *hav*. The main difference in the two processes is that, while stemming can be done simply by applying a set of rules, lemmatization requires more complex tasks such as knowing the part-of-speech of the word and understanding its context in the sentence. Given its simplicity, stemming has been applied more widely than lemmatization.

In our earlier work Flores, Moreira, and Heuser (2010), we performed a comparison between the quality of a stemming algorithm and its effectiveness in an IR system for the Portuguese language. To the best of our knowledge, this was the first investigation on the relationship between these two quality indicators. Here we expand that study by adding English, French, and Spanish to the analysis and also by performing a topic-by-topic analysis, examining in how many topics which stemmers had a better result.

We experimented with various stemmers for English, French, Portuguese, and Spanish to measure their accuracy and also to assess the gain they bring to IR. Thus, as a byproduct, this paper identifies the most accurate stemmer for each language and the one that yields the biggest IR improvement.

An important aspect, as pointed by Paice (1994), is that looking at the values for extrinsic measures does not help the designer of the stemming algorithm in seeing where the mistakes are being made. In a recent survey, Moral et al. (2014) argues that extrinsic measures (such as precision and recall) are highly dependent on other tasks within the IR pipeline, and therefore they do not provide a good solution to evaluate the quality of the stemmers independently from other processes. Intrinsic measures, on the other hand, can pinpoint more clearly where the problems are and which improvements can be made. Moreover, it is important to study the two types of measurements together to understand how one impacts the other.

The remainder of this article is organized as follows: Section 2 discusses related work and introduces some background concepts for the evaluation of stemming algorithms; Section 3 presents the experiments that measure the accuracy of the stemmers; Section 4 describes the IR experiments done to compare the impact of the stemmers over retrieval effectiveness; Section 5 investigates the correlation between both quality indicators, and Section 6 concludes the article.

2. Background and related work

Paice (1996) proposed a method to evaluate the quality of stemmers using four intrinsic metrics:

- *Overstemming Index* (OI), which calculates the number of times a stemmer mistakenly removes part of the stem as if it were part of the suffix. This type of error will typically cause unrelated words to be combined, *e.g. news* and *new* are both stemmed to *new*. OI is zero when there are no overstemming errors and one when all words are stemmed to the same stem. In IR, a high OI will potentially lead to a decrease in precision, that is, many non-relevant documents would be retrieved by the query.
- *Understemming Index* (UI), which calculates the number of times a stemmer fails to remove a suffix. This type of error will typically prevent related words from being conflated, *e.g.* if *division* is stemmed to *divis* and *divide* is stemmed to *divid*. UI is zero when there are no understemming errors and one when no words are correctly combined by the stemmer. In

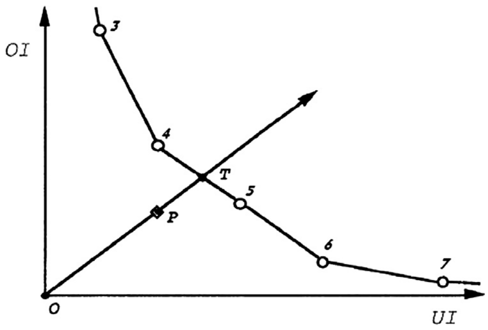


Fig. 1. ERRT computation (Paice, 1994 adapted).

IR, understemming errors will potentially lead to a decrease in recall, that is, many relevant documents will fail to be retrieved by the query.

- *Stemming Weight* (SW), which is the ratio OI / UI. A 'heavy' stemmer applies many of affix stripping rules and thus has a high OI and a low UI, while a 'light' stemmer applies few affix stripping rules and thus has a low OI and a high UI. Having a high or a low SW does not indicate whether a stemmer is better, since a better stemmer would simultaneously have a lower OI and UI.
- *Error Rate Relative to Truncation* (ERRT). The idea is that the values of (UI,OI) for a series of truncation lengths (*e.g. Trunc3* to *Trunc8*) determine a line against which the quality of the stemmer could be assessed. To exemplify, *Trunc3* is a simple stemmer that removes all but the first three letters of a word to generate the stem (*acceptance* would become *acc*). The coordinates (UI,OI) for an adequate stemmer should be below this truncation line (see Fig. 1). ERRT is obtained by extending a line from the origin through the (UI, OI) point *P* until it intersects the truncation line at *T*. ERRT is then calculated by the following formula: ERRT=length(OP) / length(OT). The lower the ERRT (*i.e.* the further from the truncation line), the better the stemmer.

Paice (1996) used his proposed method to evaluate three English stemmers: Lovins (1968), Porter (1980), and Paice (1990). He concluded that the most accurate stemmer is Paice/Husk, followed by Porter. The author also observed that Porter stemmer is lighter than Paice/Husk. The study tested word samples with different sizes (from 2654 to 9757) and concluded that the pattern of values were similar for all sample sizes. That study did not compare the effect of the stemmers on extrinsic quality metrics such as retrieval performance. Kraaij and Pohlmann (1995) employed Paice's evaluation method to assess the quality of a Dutch version of the Porter stemmer. Again this study did not test the effect of the stemmer on retrieval accuracy. For the Portuguese language, Orengo and Huyck (2001) used Paice's evaluation method to compare their proposed RSLP stemmer to the Portuguese version of the Porter stemmer. They concluded that RSLP is more accurate than Portuguese Porter. Also in Portuguese, Alvares, Garcia, and Ferraz (2005) compared STEMBR, RLSP and Portuguese Porter stemmers. The study concludes that STEMBR is heavier than Porter and RSLP and thus makes slightly fewer understemming errors than RSLP and Porter. Both works, however, have not used the ERRT metric in their evaluations. Also, none of them assesses the effects of their stemmers over retrieval performance.

There are numerous studies on the impact of stemming on the retrieval of documents in English (Harman, 1991; Hull, 1996; Krovetz, 1993). The general conclusion is that looking at the average result over a number of queries, the improvement brought by stemming is small. However, for some queries, stemming brings large gains. Savoy (2006) found statistically significant improvements in French and Portuguese using light stemmers. Other studies (Figuerola, Gómez, & de San Román, 2000; Orengo, Buriol, & Coelho, 2007; Savoy, 1999), respectively for Spanish, Portuguese, and French conclude that lighter stemmers are the best alternative. For the Czech language, Dolamic and Savoy (2009) evaluated stemming approaches based on their impact on the retrieval performance and concluded that the use of stemming was beneficial and that no statistically significant difference was found between using light and heavy stemmers for that language. These studies, however, did not evaluate the accuracy of the stemmers.

To evaluate the impact of stemming over retrieval effectiveness, the standard approach is to compare a baseline run with no stemming to a run in which stemming is used as a preprocessing step over documents and queries. The most widely used measure to assess the quality of retrieval results for a given query is the *average precision* (AvP). AvP emphasizes returning more relevant documents earlier in the ranking. When more than one query is used, the average of the AvPs, known as *Mean Average Precision* (MAP), is calculated. To compute the retrieval effectiveness measures, it is necessary to use a test collection composed by documents, queries, and relevance judgments which indicate the relevant documents for each query. Test collections do not include exhaustive relevance judgments on all query/document pairs. On a study of the impact that unjudged documents may have on the results, Voorhees (2000) concluded that very high correlations were found among the rankings of systems produced using different relevance judgment sets. This indicates that the comparative evaluation of retrieval performance is stable despite substantial differences in relevance judgments, reaffirming the use of the TREC-style collections as laboratory tools. Furthermore, evaluations of the TREC collection performed by Zobel (1998) showed that it is not biased against unjudged runs. It is also important to notice that we used at least 98 topics for each language, which can attenuate such effect even more.

3. Quality of stemming algorithms

This section describes the experimental setup and the results obtained in our analysis of stemmer accuracy referred to as intrinsic evaluation.

3.1. Experimental setup

The following stemmers were used in the experiments. Our goal was to select stemmers implemented under different strategies, *e.g.*, rule-based, statistical, and simple truncation. Some algorithms, such as Lovins and UEA-Lite, remove only one suffix per word on a longest first basis, while others may remove more than one suffix in a cascade fashion, such as Porter, Paice/Husk, RSLP and UniNE. There is also one (StemBR) that removes prefixes, in addition to suffixes. In our evaluation, *i.e.* purposely built stemmers will be called *algorithmic* stemmers (*i.e.,* all but the truncation stemmers and Stemmer-S).

- Rule-based stemmers

○ English

- Lovins: originally proposed in 1968 (Lovins, 1968), this stemmer removes endings based on the longest-match principle. It uses a large list of endings, each of which is associated with one of a number of qualitative contextual restrictions that prevent the removal of endings in certain circumstances. It was the first stemmer to be published[1].
- Porter: published in 1980 (Porter, 1980), the Porter stemmer is the most widely used stemmer for the English language and has a series of suffix stripping rules. It has already been adapted to many languages under the Snowball framework.[2]
- Paice/Husk: proposed in Paice (1990), this stemmer uses a table of rules. Each rule may specify the removal or replacement of an ending and the rules are grouped into sections corresponding to the final letter of the suffix.
- UEA-Lite: described in Jenkins and Smith (2005), it was designed to stem conservatively, recognizing words that should not be stemmed, such as proper nouns.[3]

○ French

- Porter: French version of the Porter stemmer obtained from Snowball.
- UniNE: J. Savoy from the Université de Neuchâtel has proposed stemmers for various languages.[4] This is the full version for the French language which tries to remove inflectional and derivational suffixes.
- UniNE-light: lighter version of the stemmer described above. This version only removes plural forms.
- Paice/Husk: this is a French adaptation from the original Paice/Husk algorithm.[5]

○ Portuguese

- Porter: Portuguese version of the Porter stemmer, obtained from Snowball.
- RSLP: Proposed in Orengo and Huyck (2001), it is also a suffix stripping algorithm based on rules.[6] It has about 200 stemming rules and also a list of exceptions for almost every rule.
- RSLP-S: light RSLP algorithm, using only its Plural Reduction Step, which removes, for example, the –s and –ns plural suffixes.
- UniNE: this stemmer removes inflectional suffixes for nouns and adjectives. It has many more rules than the light stemmer RSLP-S, but also fewer rules than Porter, RSLP, and StemBR.
- StemBR: Proposed in Alvares et al. (2005), it removes prefixes and suffixes.[7] The affix stripping rules were built based on the observation of the statistics of word usage on Brazilian web pages.

○ Spanish

- Porter: Spanish version of the Porter stemmer, available from Snowball.
- UniNE: Spanish version of the UniNE stemmer.

The stemmers below are language independent and were applied to all four languages.

- Statistical Stemmers:
- Linguistica: a statistical stemmer,[8] which is language independent, described in Oard, Levow, and Cabezas (2001). We used Linguistica version 3.2.6 for Windows. It is important to state that Linguistica can only work, theoretically, with at most 10,000,000 tokens (words). However, our IR corpora are larger than that. Furthermore, in our tests, it could only

[1] The implementation of the Lovins stemmer we used in the experiments was obtained from http://www.cs.waikato.ac.nz/~eibe/stemmers/index.html.

[2] http://snowball.tartarus.org/.

[3] The implementation of the UEA stemmer was obtained from http://www2.cmp.uea.ac.uk/~djs/projects/UEAlite/stemmer.html.

[4] http://members.unine.ch/jacques.savoy/clef/index.html.

[5] The implementation of the Paice/Husk algorithm for French was obtained from http://alx2002.free.fr/utilitarism/stemmer/stemmer_fr.html.

[6] The implementation of the RSLP algorithm was obtained from http://www.inf.ufrgs.br/~arcoelho/rslp/integrando_rslp.html.

[7] The implementation of the StemBR algorithm was obtained directly from the author.

[8] The Linguistica software was obtained from http://linguistica.uchicago.edu/downloads.html.

Table 1
Sample output from the different stemmers to the same input text.

NoStem	saby had complained unsuccessfully on wednesday that vatanen and his citroen team were gaining an unfair advantage from radio contact with team manager guy frequelin in his helicopter.
Lovins	sab had complain unsucces on wednesda that vatan and hi citro team wer gain an unfair advant from radi contact with team manager guy frequelin in hi helicopter.
Paice/Husk	saby had complain unsuccess on wednesday that vatan and his citro team were gain an unfair adv from radio contact wi team manag guy frequelin in his helicopt.
Porter	sabi had complain unsuccess on wednesday that vatanen and his citroen team were gain an unfair advantag from radio contact with team manag guy frequelin in his helicopt.
UEA-Lite	saby had complain unsuccessfully on wednesday that vatanen and he citroen team were gain an unfair advantage from radio contact with team manager guy frequelin in he helicopter.
Linguistica	saby had complain unsuccessful on wednesday t vatanen an his citroen team were gain an unfair advantage from radio contact with team manag guy frequelin in hi helicopt.
Stemmer-S	saby had complained unsuccessfully on wednesday that vatanen and hi citroen team were gaining an unfair advantage from radio contact with team manager guy frequelin in hi helicopter.
GRAS	saby had complain unsuccessful on wednesday that vatanen and his citroen team were gaining an unfair advantage from radio contact with team manager guy frequelin in his helicopter
Trunc4	saby had comp unsu on wedn that vata and his citr team were gain an unfa adva from radi cont with team mana guy freq in his heli.

work for corpora of up to 20MB. Therefore, for each language, we assembled a sample corpus with randomly picked articles, along with the 2000 words from the Paice experiment. Later, we used the ‘stems file’ that Linguistica produced as if they were the stemming rules for each language to process the entire corpus.

- GRAS (graph-based stemmer): proposed by Paik et al. (2011), Paik, Pal, and Parui (2011) it is based on word co-occurrences. The process involves the creation of a graph in which the word variants are the nodes and, whenever they co-occur, an edge is drawn between them. The edges are stronger the more the two words appear together. The parameters used in our experiment are the same as recommended by its authors, namely, $\alpha = 4$ and $\delta = 0.8$.

- Stemmer-S: a simple light stemming algorithm, implemented by us, which only removes the finals of the words. It is important to notice that this algorithm makes sense as an approximate plural reduction stemmer for the four languages we are testing, but this would not work for all languages. In Germanic languages, for example, a few letters and diacritics are used to denote the plural form.
- Truncation Stemmers
  - TruncN: a simple truncation algorithm, which simply removes all but the first n letters of a word. We have tested with n varying from three to eight. They will be referenced as Trunc3, Trunc4, Trunc5, Trunc6, Trunc7, and Trunc8.

In addition, to serve as a baseline, runs in which no stemming was used are identified as *NoStem*.

Table 1 shows the output of each stemmer with the same text as input for the English language, to give an idea of how stemming is done. We can see that some stemmers are more aggressive than others in removing affixes.

The computation of the quality metrics proposed by Paice requires groups of semantically and morphologically related words. In order to generate such groups, we obtained a list of words available from Snowball for the Portuguese, Spanish, and French languages and selected a sample of words in each of them. The goal while selecting the sample was to have words with diverse starting characters and also to have groups of different sizes. The sample was manually divided into groups of semantically and morphologically related words. For the English language, we took a sample that was already divided. [9] The English sample has 1741 words, divided into 665 groups. The French sample has 2001 words, divided into 442 groups. The Portuguese sample has 2854 words, divided into 888 groups. Finally, the Spanish sample has 1752 words, divided into 604 groups.

It is worth mentioning that we also performed the experiments in this Section (and the correlation experiments in Section 5) with different sets of words. The first was a sample of 2000 words randomly selected from the corpora used in the IR tests of Section 4. This was done in order to assess whether the Porter stemmers might have had an unfair advantage due to us having used a sample of words obtained from Snowball. Additionally, we used other two different set of words for each language: 500 and 1000 words randomly selected from the original words (around 2000) used in our experiments. This was done in order to test whether the selection of terms affects the intrinsic measures. The results obtained with these new samples were similar to those obtained with the original ones, that is, the correlations between extrinsic and intrinsic metrics remained alike and the ranking of stemmers with regards to Paice’s metrics were similar. Due to space restrictions, we only show and discuss the results obtained with the original samples. However, full results on the other samples can be accessed at www.inf.ufrgs.br/∼fnflores/other_samples.

We developed a tool to compute Paice’s evaluation method. The tool calculates all four quality metrics, including ERRT, which involves computing overstemming and understemming indices for different truncation algorithms and constructing the truncation line by connecting the coordinates (UI, OI) for each truncation length (see Fig. 1). Then, the tool calculates

[9] The list of words separated into groups for English was originally obtained from the Lancaster University’s website in a page with resources for the Paice test for stemming algorithms, which no longer exists. We have made it available at http://www.inf.ufrgs.br/∼fnflores/english_words.txt.

Table 2
Result of Paice's evaluation method.

	UI	OI	SW	ERRT		UI	OI	SW	ERRT
English					**French**				
Lovins	0.2753	0.0016	0.0059	1.5711	Paice/Husk	**0.1650**	0.0001	0.0007	**0.5131**
Paice/Husk	**0.2275**	0.0004	0.0019	0.8579	Porter	0.2658	0.0001	0.0003	0.6429
Porter	0.3032	0.0001	0.0003	**0.6256**	UniNE-Light	0.8396	**0.0000**	0.0000	0.9806
UEA-Lite	0.5471	**0.0000**	0.0000	0.7702	UniNE	0.7803	**0.0000**	0.0000	1.0193
Linguistica	0.8163	0.0001	0.0001	1.2038	Linguistica	0.4352	0.0001	0.0002	0.8201
Stemmer-S	0.8151	**0.0000**	0.0000	0.9239	Stemmer-S	0.9525	**0.0000**	0.0000	0.9758
GRAS	0.5135	0.0001	0.0002	0.8789	GRAS	0.4640	0.0001	0.0001	0.7720
Trunc3	0.0013	0.0063	4.8652	1.0000	Trunc3	0.0049	0.0062	1.2585	1.0000
Trunc4	0.0989	0.0015	0.0150	1.0000	Trunc4	0.0255	0.0012	0.0460	1.0000
Trunc5	0.2882	0.0004	0.0013	1.0000	Trunc5	0.1741	0.0004	0.0021	1.0000
Trunc6	0.5019	0.0001	0.0003	1.0000	Trunc6	0.4397	0.0001	0.0003	1.0000
Trunc7	0.7402	0.0000	0.0000	1.0000	Trunc7	0.7098	0.0000	0.0001	1.0000
Trunc8	0.8951	0.0000	0.0000	1.0000	Trunc8	0.8514	0.0000	0.0000	1.0000
NoStem	1.0000	0.0000	0.0000	1.0000	NoStem	1.0000	0.0000	0.0000	1.0000
Portuguese					**Spanish**				
Porter	0.3015	0.0001	0.0005	0.7159	Porter	**0.5544**	0.0002	0.0004	**0.9513**
RSLP	**0.1905**	0.0003	0.0014	**0.5691**	UniNE	0.9085	**0.0000**	0.0000	1.0701
RSLP-S	0.9515	**0.0000**	0.0000	0.9959	Linguistica	0.8738	0.0001	0.0001	1.1698
UniNE	0.8863	**0.0000**	0.0000	1.1301	Stemmer-S	0.9566	**0.0000**	0.0000	1.0318
StemBR	0.2669	0.0003	0.0012	0.7650	GRAS	0.7057	0.0002	0.0003	1.1004
Linguistica	0.7880	**0.0000**	0.0000	1.1825	Trunc3	0.0707	0.0270	0.3815	1.0000
Stemmer-S	0.9580	**0.0000**	0.0000	1.0027	Trunc4	0.1587	0.0036	0.0228	1.0000
GRAS	0.5577	**0.0000**	0.0001	0.9371	Trunc5	0.2930	0.0009	0.0030	1.0000
Trunc3	0.0305	0.0158	0.5184	1.0000	Trunc6	0.5244	0.0003	0.0005	1.0000
Trunc4	0.1079	0.0030	0.0277	1.0000	Trunc7	0.7204	0.0001	0.0001	1.0000
Trunc5	0.2676	0.0007	0.0025	1.0000	Trunc8	0.8221	0.0000	0.0001	1.0000
Trunc6	0.4521	0.0001	0.0002	1.0000	NoStem	1.0000	0.0000	0.0000	1.0000
Trunc7	0.6484	0.0000	0.0000	1.0000					
Trunc8	0.8150	0.0000	0.0000	1.0000					
NoStem	1.0000	0.0000	0.0000	1.0000					

the distance between the coordinate (UI, OI) obtained from the stemming algorithm and the truncation line. This implementation, which is language-independent, *i.e.*, it can be used to evaluate stemming algorithms of any language, can be freely accessed at http://www.inf.ufrgs.br/~fnflores/paice_tool.

3.2. Results

In this section, we show the results obtained using Paice's evaluation method for all stemmers in all four languages. Table 2 shows the figures for all four intrinsic quality metrics. The best results for each metric are in bold.

In terms of UI, as expected in all languages, the best result is obtained by Trunc3 as it rarely fails to conflate related forms of words. On the other hand, Trunc3 has the highest OI. The best UI of an algorithmic stemmer, in English and French, is achieved by Paice/Husk. In Portuguese, the best UI is RSLP, while for Spanish it is Porter.

The algorithmic stemmers which make the fewest overstemming errors (*i.e.*, smallest OI) are UEA-Lite for English; UniNE and UniNE-Ligh for French; RSLP-S, UniNE, Linguistica, and GRAS for Portuguese; and UniNE for Spanish. In all languages, Stemmer-S also had the lowest OI. These results are expected, since those stemmers have fewer suffix stripping rules. Recall that UI and OI alone do not enable the identification of the best algorithms.

For English, the best stemmer in terms of ERRT is Porter, followed by UEA-Lite and Paice/Husk. Lovins has a very bad ERRT, caused by its very high OI rate. In terms of SW, UEA-Lite is the lightest, and Lovins is the heaviest, followed by Paice/Husk. This result corroborates the findings by Paice (1994).

In French, the best stemming algorithm is Paice/Husk, according to ERRT, followed by Porter. UniNE for French and UniNE-Light had the worst results, because of the high UI values. They are also the lightest, according to SW.

In terms of ERRT, for the Portuguese language, the best stemmers are, in order, RSLP, Porter for Portuguese, StemBR and RSLP-S, with the fourth well behind the first three. We can also notice that RSLP and StemBR are heavier than Porter for Portuguese and UniNE for Portuguese, according to SW. This fact was also found in Alvares et al. (2005) and Orengo and Huyck (2001). Besides, even though RSLP has a similar SW to StemBR, the first outperforms the last in both UI and OI. However, RSLP outperforms Porter for Portuguese and UniNE for Portuguese only in UI, since Porter and UniNE have better OIs. Finally, we can also see that the light stemmers, RSLP-S and Stemmer-S, had very bad results in ERRT, being very close to 1. This happens because their UI is very high, also close to 1, since they have very few affix stripping rules and therefore only a small number of related words end up with the same stem. Linguistica for Portuguese and Spanish had the worst ERRT for these languages.

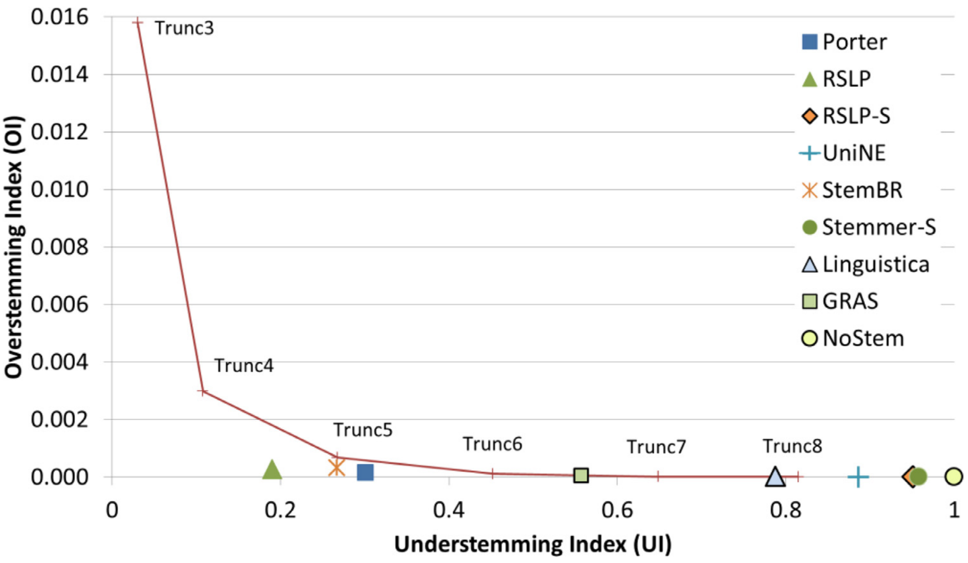


Fig. 2. ERRT Plot for the Portuguese Stemmers.

Table 3
Details of the test collections.

	English	French	Portuguese	Spanish
Documents	169,477	129,806	103,913	454,045
Queries	176 (1–200)	185 (1–200)	98 (251–350)	156 (41–200)
Relevant docs	3262	4069	2159	7916
Index terms	595,025	558,048	341,362	1,429,868
Size	213MB	113MB	86MB	371MB

In Spanish, Porter has the best result according to ERRT and is the heaviest. It is interesting to notice that, for English, the best stemmers in terms of ERRT are the lightest, while the opposite happens for the remaining languages.

Finally, ERRT is exactly 1 for all truncation stemmers (since they are, by definition, on the truncation line) and for NoStem, since it can be seen as a TruncN algorithm, with *n* being the length of the longest word in the sample.

Fig. 2 shows a graphic interpretation of ERRT, for the Portuguese stemmers. Recall that a good stemmer should be below the truncation line, and as far as possible from it. Thus, RSLP can be considered the best stemmer, followed by Portuguese Porter and StemBR. The light stemmers RSLP-S and Stemmer-S, as well as the medium weight stemmer UniNE, are close to or above the truncation line and therefore can be considered worse. The same happens with Linguistica. We omitted the plots for other languages for space reasons but they are similar: the stemmers with ERRT lower than 1 are below the truncation line, while the ones with ERRT greater than 1 are above the line.

4. Stemmers and Information Retrieval

This section presents the experiments we carried out in order to assess how the different stemmers impact retrieval effectiveness, referred to as extrinsic evaluation.

4.1. Experimental setup

In this experiment, we used the data collections from the CLEF campaigns (Peters et al., 2007 , 2006; Peters, Braschler, & Clough, 2012) of 2005 and 2006 (obtained from ELDA[10]). The documents are news articles from Glasgow Herald and LA Times, for English; Le Monde and SDA, for French; Folha de São Paulo, for Portuguese; and Agencia EFE, for Spanish. Queries with no relevant documents in the collection were not considered, and that is why we have a number of queries lower than the subtraction of the numbers of the last and the first topics. Details are in Table 3.

To index the data collection, we used the Zettair Search Engine (Billerbeck et al., 2004). Before indexing the collections, we processed each word in the documents and in the topics with the stemmers listed in Section 3.1, and then used these stemmed files as input to Zettair. The similarity metric used to rank documents in response to queries was Okapi BM25.

[10] http://www.elra.info/.

Table 4
Results of the retrieval experiment.

	Distinct index terms	MAP		Distinct index terms	MAP
English			**French**		
Lovins	537,398 (−9.68%)	0.3095	Paice/Husk	**487,604 (−12.62%)**	0.2670
Paice/Husk	**286,481 (−51.85%)**	0.2244	Porter	497,840 (−10.79%)	0.2703
Porter	560,771 (−5.76%)	0.3135	UniNE-Light	543,869 (−2.54%)	0.2538
UEA-Lite	552,975 (−7.07%)	0.2624	UniNE	530,651 (−4.91%)	**0.2735**
Linguistica	576,695 (−3.08%)	0.2405	Linguistica	548,865 (−1.65%)	0.2259
Stemmer-S	557,590 – 6.29%)	0.3243	Stemmer-S	538,324 (−3.53%)	0.2682
GRAS	564,599 (−5.11%)	0.3292	GRAS	526,476 (−5.65%)	0.2679
Trunc3	**275,519 (−53.70%)**	0.1678	Trunc3	**94,527 (−83.06%)**	0.1326
Trunc4	347,741 (−41.56%)	0.2939	Trunc4	352,939 (−36.75%)	0.2496
Trunc5	438,214 (−26.35%)	0.3185	Trunc5	413,285 (−25.94%)	0.2712
Trunc6	509,305 (−14.41%)	**0.3319**	Trunc6	473,629 (−15.13%)	0.2702
Trunc7	549,558 (−7.64%)	0.3291	Trunc7	502,335 (−9.98%)	0.2664
Trunc8	570,538 (−4.12%)	0.3160	Trunc8	521,734 (−6.51%)	0.2515
NoStem	595,025	0.3037	NoStem	558,048	0.2383
Portuguese			**Spanish**		
Porter	250,864 (−26.51%)	0.2981	Porter	**1,294,543 (−9.46%)**	0.3071
RSLP	237,490 (−30.43%)	0.2898	UniNE	1,394,913 (−2.44%)	**0.3172**
RSLP-S	313,737 (−8.09%)	0.2938	Linguistica	1,353,478 (−5.34%)	0.2242
UniNE	306,836 (−10.11%)	**0.2984**	Stemmer-S	1,409,884 (−1.40%)	0.2980
StemBR	**233,536 (−31.59%)**	0.2829	GRAS	1,345,456 (−5.90%)	0.3153
Linguistica	325,517 (−4.64%)	0.2711	Trunc3	**381,516 (−73.32%)**	0.1984
Stemmer-S	316,753 (−7.21%)	0.2736	Trunc4	468,346 (−67.25%)	0.2753
GRAS	312,430 (−8.47%)	0.2684	Trunc5	644,163 (−54.95%)	0.3110
Trunc3	**72,750 (−77.36%)**	0.1784	Trunc6	727,834 (−49.10%)	0.3155
Trunc4	109,093 (−68.04%)	0.2706	Trunc7	938,299 (−34.38%)	0.3035
Trunc5	163,068 (−52.23%)	0.2957	Trunc8	1,261,082 (−11.80%)	0.2952
Trunc6	212,723 (−37.68%)	0.2751	NoStem	1,429,868	0.2710
Trunc7	250,782 (−26.53%)	0.2812			
Trunc8	280,803 (−17.74%)	0.2718			
NoStem	341,362	0.2587			

4.2. Results

In this section, we show the results of mean average precision (MAP) and number of distinct terms indexed for each stemmer. The results are summarized on Table 4. The second column in Table 4 shows the number of distinct index terms after Zettair indexed the collection. Trunc3 had the best score here for all languages, since one of the goals of stemming is to reduce the number of index entries. The largest reduction rate among the stemmers was obtained by Paice/Husk, in English and French; StemBR, in Portuguese; and Porter, in Spanish. It is interesting to notice that, since English words tend to be shorter, English has the smallest reduction for Trunc3 (around 50%) and Trunc8 (around 4%).

The third column of Table 4 shows the results for MAP. We can see that, in Portuguese, Spanish, and French, the UniNE stemmer achieved the best result in improving IR effectiveness, while Trunc3 was the worst for all languages. In English, the best result in improving IR effectiveness was surprisingly not achieved by an algorithmic stemmer, but by a truncation algorithm (Trunc6). This finding is in line with McNamee, Nicholas, and Mayfield (2009), who found that truncation algorithms with at most 5 characters were among the top performers over a number of truncation methods. Our results seem to indicate that the hypothesis that stemming in English is not as beneficial as for the Romance languages is valid, since the latter have more variant forms in words that a stemmer could explore.

We also performed two-tailed paired *t-tests*, to evaluate whether the differences in MAP were statistically significant. We used $\alpha = 0.05$ as threshold for statistical significance.

Table 5 shows the results obtained, with "=" meaning that no statistical significance was found, "+" meaning that the difference was significant and the stemmer of the row was better than the one of the column and "–" meaning the opposite (column better than row).

By analyzing the results of the *t-tests*, we can see that only UniNE (in Portuguese, French and Spanish), Porter (in Portuguese, French and Spanish – but not in English), RSLP, RSLP-S, Paice/Husk (in French, but not in English), GRAS (in all languages but Portuguese), Stemmer-S (in all languages), Trunc5 (in Portuguese, French, and Spanish), Trunc6 (in French, Spanish, and English), Trunc7 (in all languages) and Trunc8 (in French, Spanish, and English) have shown statistically significant improvements in relation to NoStem. Trunc3 was significantly worse than all other stemmers, including NoStem, in all languages.

For English, the only algorithmic stemmer that achieved a statistically significant improvement in relation to NoStem was GRAS. Paice/Husk and UEA-Lite were actually significantly worse. Besides, in Portuguese, StemBR was significantly worse than UniNE, Porter, and RSLP, while RSLP-S was significantly better than Stemmer-S and Trunc8. It is also interesting to

Table 5
Results of the *t*-test per language.

(a) English

	Lovins	Paice/Husk	Porter	UEA-Lite	Linguistica	GRAS	Stemmer-S	Trunc3	Trunc4	Trunc5	Trunc6	Trunc7	Trunc8
NoStem	=	+	=	+	+	–	–	+	=	=	–	–	–
Lovins		+	=	+	+	=	=	+	=	=	=	=	=
Paice/Husk			–	=	=	–	–	+	–	–	–	–	–
Porter				+	+	=	=	+	=	=	=	=	=
UEA-Lite					+	–	–	+	–	–	–	–	–
Linguistica						–	–	+	–	–	–	–	–
GRAS							=	+	+	=	=	=	=
Stemmer-S								+	+	=	=	=	=
Trunc3									–	–	–	–	–
Trunc4										–	–	–	=
Trunc5											–	=	=
Trunc6												=	=
Trunc7													=

(b) French

	Paice/Husk	Porter	UniNE-Light	UniNE	Linguistica	GRAS	Stemmer-S	Trunc3	Trunc4	Trunc5	Trunc6	Trunc7	Trunc8
NoStem	–	–	=	–	=	–	–	+	=	–	–	–	–
Paice/Husk		=	=	=	+	=	=	+	=	=	=	=	=
Porter			=	=	+	=	=	+	=	=	=	=	=
UniNE-Light				–	+	–	–	+	=	=	–	=	=
UniNE					+	=	=	+	+	=	=	=	=
Linguistica						–	–	+	=	–	–	–	–
GRAS							=	+	+	=	=	=	+
Stemmer-S								+	=	=	=	=	=
Trunc3									–	–	–	–	–
Trunc4										–	–	=	=
Trunc5											=	=	=
Trunc6												=	=
Trunc7													+

(c) Portuguese

	Porter	RSLP	RSLP-S	StemBR	UniNE	Linguistica	GRAS	Stemmer-S	Trunc3	Trunc4	Trunc5	Trunc6	Trunc7	Trunc8
NoStem	–	–	–	=	–	=	=	–	+	=	–	=	–	=
Porter		=	=	+	=	+	+	=	+	=	=	=	=	=
RSLP			=	+	=	=	=	=	+	=	=	=	=	=
RSLP-S				=	=	+	+	+	+	=	=	=	=	+
StemBR					–	=	=	=	+	=	=	=	=	=
UniNe						+	+	=	+	=	=	=	=	+
Linguistica							=	=	+	=	=	=	=	=
GRAS								=	+	=	=	=	=	=
Stemmer-S									+	=	=	=	=	=
Trunc3										–	–	–	–	–
Trunc4											=	=	=	=
Trunc5												=	=	=
Trunc6													=	=
Trunc7														=

(d) Spanish

	Porter	UniNE	Linguistica	GRAS	Stemmer-S	Trunc3	Trunc4	Trunc5	Trunc6	Trunc7	Trunc8
NoStem	–	–	+	–	–	+	=	–	–	–	–
Porter		=	+	=	=	+	+	=	=	=	=
UniNE			+	=	+	+	+	=	=	=	+
Linguistica				–	–	=	–	–	–	–	–
GRAS					+	+	+	=	=	=	+
Stemmer-S						+	=	=	=	=	=
Trunc3							–	–	–	–	–
Trunc4								–	–	–	=
Trunc5									=	=	=
Trunc6										=	=
Trunc7											=

Table 6
Number of topics in which stemming was beneficial.

	English (%)	French (%)	Portuguese (%)	**Spanish (%)**
Topics in which a stemmer was better than NoStem	71.59	67.03	79.59	75.00
Topics in which no stemmer was better than NoStem	28.41	32.97	20.41	25.00

note that UniNE-Light was significantly worse than UniNE and Stemmer-S, for French. Furthermore, UniNE, in Spanish, was significantly better than Stemmer-S.

Comparing statistical and rule-based stemmers, McNamee, Nicholas, and Mayfield (2008) concluded that unsupervised (statistical) stemmers outperform baseline runs in which no stemming is used and that for low complexity languages (such as Romance languages), rule-based stemmers are better. In short, statistical stemming is better than no stemming and, when available, rule-based stemmers tend to be superior. Our findings partially corroborate theirs as GRAS achieved significant improvements in all languages but Portuguese, having the best results in English. On the other hand, Linguistica was significantly worse than NoStem in English and Spanish and showed no significant difference in Portuguese and French.

4.3. Topic-by-topic analysis

In this section, we perform a topic-by-topic analysis of the IR experiment considering each individual AvP (average precision), rather than looking only at the mean (MAP). Fig. 3 shows, for each stemmer, in how many topics it was better/worse/equivalent to NoStem. Fig. 4 shows the difference between the topics that each stemmer was better and the ones they were worse than NoStem. We considered that a stemmer had the same result as NoStem in a topic if the proportional difference between their AvPs was less than 5%. The choice of this percentage is consistent with Buckley and Voorhees (2000), who state that a difference greater than 5% in AvP is significant.

While Fig. 4 shows the magnitude of the gain/loss for all topics, Fig. 3 shows the number of topics with gains, losses, and ties. In both figures, the columns are sorted in decreasing order and the top scoring stemmers are quite different between them. For example, in English, the stemmer which the biggest number of topics with improvements in comparison to NoStem was Paice/Husk (Fig. 3a). However, the same stemmer also had the most topics with losses and its overall result was negative (as shown in Fig. 4a). The only stemmer which achieved the best results in both cases was RSLP (Portuguese), however it did not get the highest MAP (as shown in Table 3). With the exception of Spanish, the stemmer with the best MAP was different from the stemmer with the biggest difference in percentage. A curious case was found in Trunc7 (French). Even though its better-worse difference was only 2.7%, the *t-test* showed that it was significantly better than NoStem in terms of MAP.

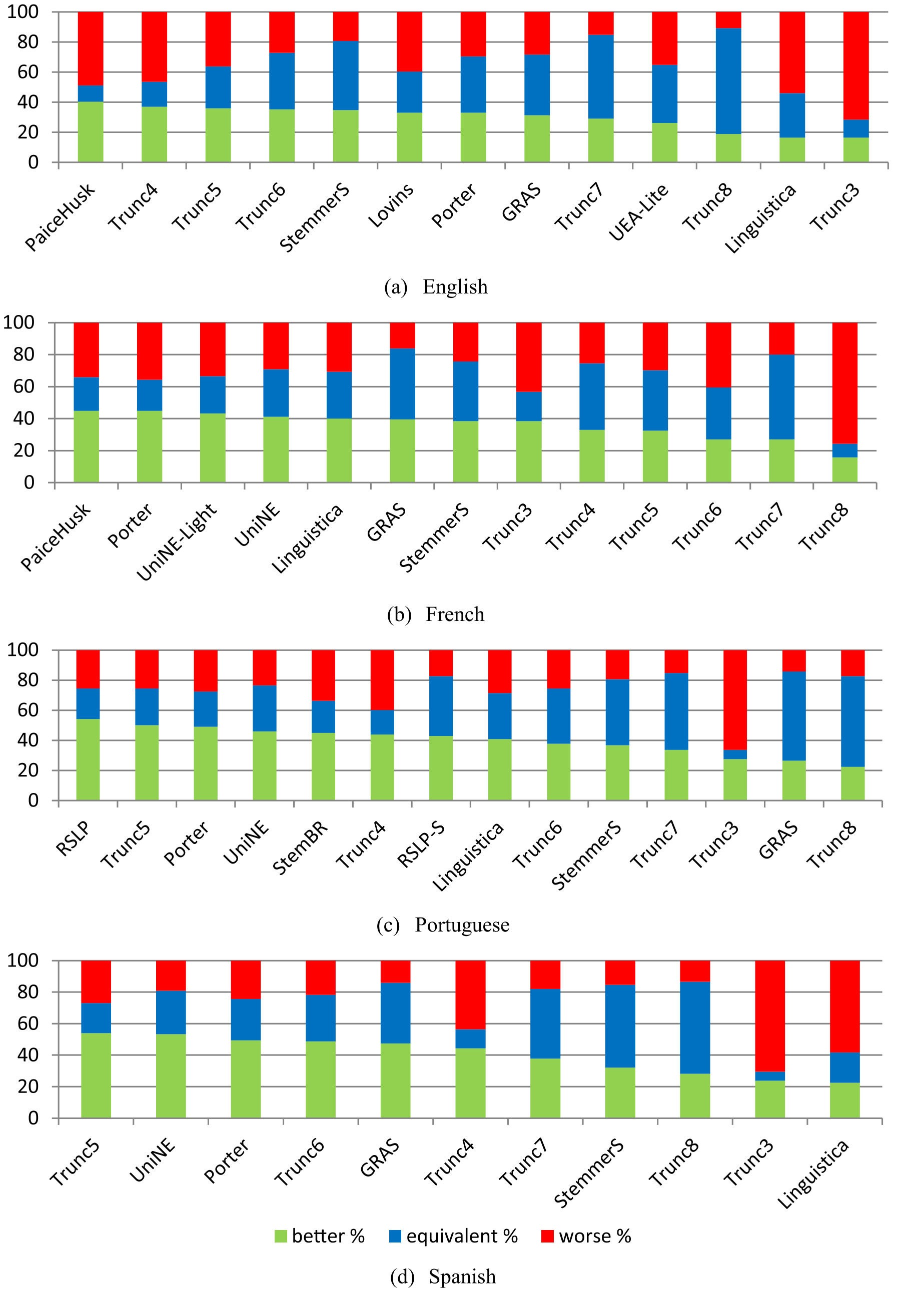
In this topic-by-topic analysis, we also computed the number of topics in which stemming was beneficial (*i.e.*, improvement in average precision). The results are summarized in Table 6. These results show that the smallest number of topics in which a stemmer was better than NoStem was French, with 67.03%. However, French had good results in MAP, since 9 out of the 12 stemmers that were tested performed statistically significant better than NoStem. This can be explained by the fact that, for the topics that were improved, the improvement was large. Also, in the topics in which retrieval quality was hurt by stemming, the loss was small.

Finally, we give examples of topics in which the results were improved and hurt by stemming. Topic 67 (in English) on *ship collisions* had a very bad average precision when no stemming was used (about 0.01) as the relevant documents contained only the singular form *collision*. All stemmers were able to reduce the plural and, as a result, average precision was as high as 0.58 in the stemmed runs. In contrast, for topic 152 on *Children's Rights* the reduction of *rights* to *right* introduced noise into the query which ended up retrieving many irrelevant documents. As a result, its average precision, which was 0.30 without stemming, dropped as low as 0.0005 in one of the stemmed runs.

5. Correlation between Stemming Accuracy and IR performance

Recall that stemming aims at solving the vocabulary mismatch problem which is known to be one of the core issues in IR. The perfect stemmer should be able to conflate all variant forms of the words to a single stem (*i.e.*, no understemming), and at the same time, avoid conflating unrelated words to the same stem (*i.e.*, no overstemming). In this sense, the intuition is that the quality of the stemmer (as measured intrinsically through Paice's metrics) should directly impact the quality of the retrieval results (as measured in terms of mean average precision). Intuitively these two metrics should be correlated, thus, in this section, we analyze the correlations between Stemming Accuracy (as calculated by Paice's evaluation method) and IR's retrieval performance measured by MAP and number of distinct index terms. We used Pearson product-moment correlation coefficient (Pearson, 1895). The results are summarized on Table 7. An asterisk '*' means that the correlation is statistically significant, using 0.05 as the threshold of significance of the two-tailed probability values.

We found the assumption that there would be a strong negative correlation between the errors made by the stemmer and the measures for IR performance such as MAP does not hold. The correlation between ERRT and MAP is weak for all four languages (and, in some cases, the correlation is even positive).



(a) English

(b) French

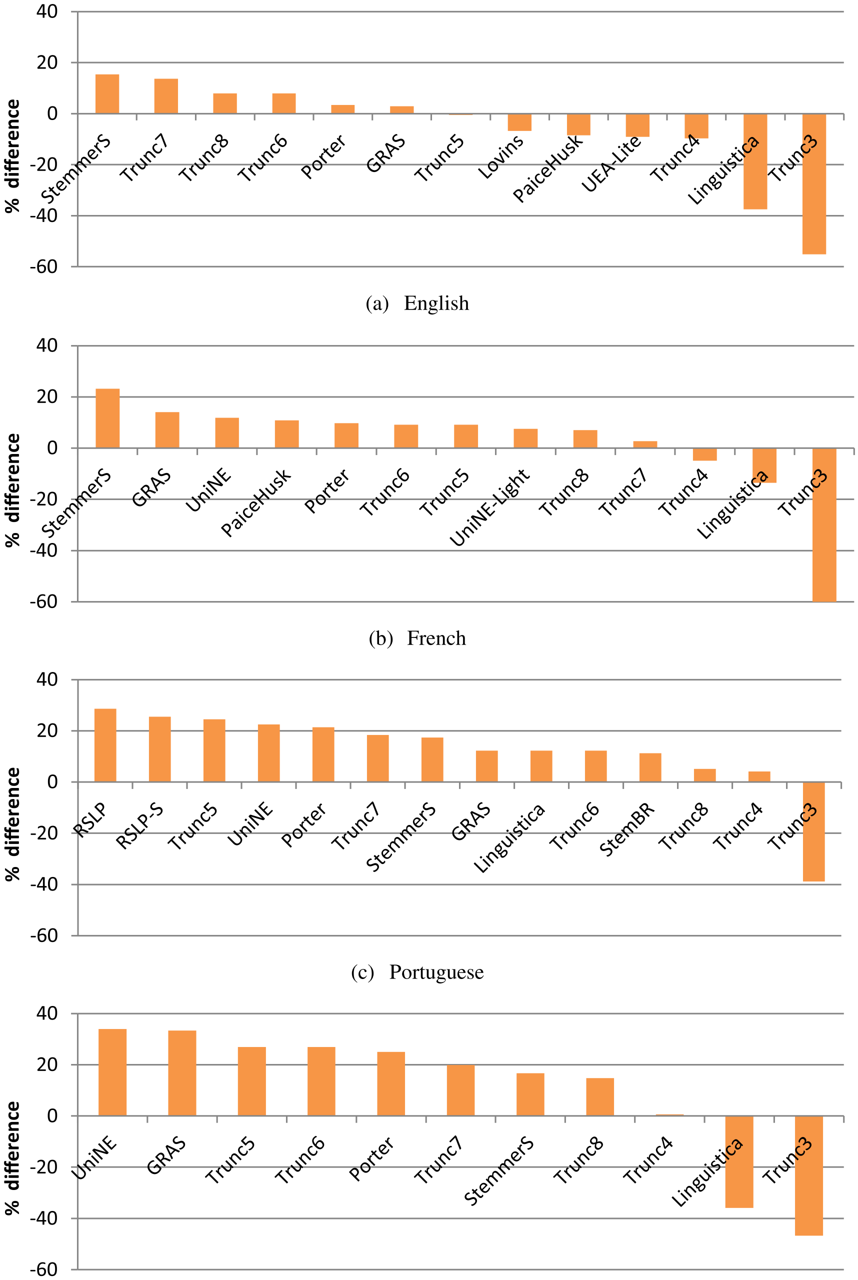
(c) Portuguese

(d) Spanish

Fig. 3. Percentage of topics which had better, equivalent, or worse results compared to NoStem. The bars are sorted in descending order of the number of topics in which gains were found.

Analyzing the results of the experiments from Sections 3 and 4, we observed that a stemmer that has lower ERRT is not necessarily better for IR than another stemmer with a higher ERRT. This is especially true in cases of light stemming, like RSLP-S, UniNE (all languages) and Stemmer-S (all languages), which had very good performances in IR, but an ERRT close to one. Remarkably, UniNE stemmers had the highest ERRT in French and second highest in Portuguese and Spanish, but also the highest MAP, with statistically significant improvements in relation to NoStem.

ERRT computes the distance from the truncation line, assuming that a truncation stemming algorithm is bad and should be avoided. Paice (1994) states that length truncation is the crudest method of stemming and that we would obviously expect any rule-based or table-based stemmer to do better. However, that turned out to be false, as Trunc5, Trunc6, Trunc7 and Trunc8 had statistically significant improvements in relation to NoStem in at least three different languages (Trunc7 in all four).



(a) English

(b) French

(c) Portuguese

(d) Spanish

Fig. 4. Results per topic (difference in %) in relation to NoStem.

Nevertheless, ERRT was able to distinguish the quality of stemmers of similar weight, but only in Portuguese. We can see that, for the light stemmers, RSLP-S was significantly better in MAP than Stemmer-S and had lower ERRT as well. Furthermore, for the heavy stemmers, StemBR had higher ERRT than Porter and RSLP, and also performed significantly worse in terms of MAP. RSLP, however, was much better than Porter in ERRT, but they did not have a significant difference in terms of MAP.

Notwithstanding, this distinction did not occur in the other languages. In Spanish, UniNE was significantly better than Stemmer-S, and both are light, but UniNE had a worse ERRT. In French, UniNE-Light was significantly worse than UniNE, and both are light, but UniNE-Light had better ERRT. Finally, in English, UEA-Lite was significantly worse than Stemmer-S, and both are light, but UEA-Lite had better ERRT.

Table 7
Correlation between Stemming Accuracy (OI, UI, SW, ERRT) and IR performance (MAP and distinct index terms).

	MAP	Distinct index terms
English		
Overstemming index	−0.70*	−0.67*
Understemming index	0.42	0.79*
Stemming weight	−0.72*	−0.57*
ERRT	0.00	0.06
French		
Overstemming index	−0.91*	−0.95*
Understemming index	0.33	0.69*
Stemming weight	−0.93*	−0.90*
ERRT	−0.19	−0.18
Portuguese		
Overstemming index	−0.90*	−0,71*
Understemming index	0.33	0.87*
Stemming weight	−0.91*	−0.63*
ERRT	−0.17	0.14
Spanish		
Overstemming index	−0.73*	−0.61*
Understemming index	0.29	0.93*
Stemming weight	−0.73*	−0.57
ERRT	−0.24	0.40

The statistically significant negative correlations that were found happened between MAP and OI, and MAP and SW, in all languages, especially French and Portuguese. In English, the correlation was the weakest, but also significant. This fact indicates that, for IR purposes, overstemming errors are more serious than understemming errors. As shown in Table 7, UI is positively correlated with MAP in all languages, even though this correlation is weak. These conclusions corroborate previous studies (Figuerola et al., 2000; Savoy, 2006) which advocate the use of lighter stemming alternatives for IR.

Finally, analyzing the third column, which shows the correlation with the number of distinct index terms, the results were as expected. The correlation with UI was positively strong, while the correlation with OI and SW was negative, for all languages. This demonstrates that the heavier the stemmer, the greater number of words which will be conflated to the same stem and, therefore, the number of distinct index terms will be reduced. The correlation with ERRT was very weak for all languages, which shows that these two measures are not related.

6. Conclusion

This article evaluates several Portuguese, Spanish, French, and English stemmers found in the literature in terms of their accuracy and in terms of their impact on retrieval effectiveness. Our aim was to assess if the quality of a stemming algorithm, particularly the quality measured by Paice's evaluation method, would translate into IR improvement and to analyze the differences on this matter for the different languages. Our conclusion was that the most accurate stemmer was not the one to achieve the greatest improvement in IR, in none of the languages.

We have implemented Paice's evaluation method and made the tool available. With our implementation and a sample of words, it becomes very simple and fast to analyze stemmers for any language according to Paice's evaluation method. The tool calculates UI, OI, SW, and ERRT. If one does not have any previous information about the stemmers to be tested, Paice's evaluation method, with the SW metric, correctly shows which ones are light and which ones are heavy stemmers. In Portuguese, it also correctly spotted the better stemmers when they had similar weights but significant difference in quality, using ERRT.

In relation to the different languages, the results have shown that, for English, stemming is not as beneficial in terms of IR as for the three Romance languages. Even though there are very accurate English stemmers, with good results in terms of UI, OI and ERRT, no algorithmic stemmer was able to have statistically significant improvements in MAP, with the best result coming from the Trunc6 algorithm. However, the experiments have shown that more than 70% of the query topics in English had an improvement in AvP, which is a similar percentage to the other languages.

Additionally, experiments also showed that light stemming is a good approach to improve MAP. Stemmer-S and Trunc7, which are very simple, have had statistically significant improvements in all languages. To reduce the number of distinct index terms, however, a heavier form of stemming is still needed.

Further studies can be done to see if the same correlations found in this article will apply to other languages with different characteristics, such as Indian dialects or the Arabic language. Fattah, Ren, and Kuroiwa (2006), for example, show that we have different challenges when building a stemmer for the Arabic language. Moreover, it could be investigated if we can create a new intrinsic metric that is highly correlated to MAP and other extrinsic metrics. As recently there has been a lot of work on building stemmers for languages with scarce resources, such as Saharia, Sharma, and Kalita (2014) and

Ramachandran and Krishnamurthi (2012), the development of such a metric would be even more meaningful, because we would not need to have large document collections, topics and relevance judgments to evaluate these new stemmers.

Additionally, further studies can use our implementation of Paice's evaluation method and the conclusions we obtained here to improve existing stemming algorithms or new stemmers to be created. Finally, one could investigate the query topics in which stemming is being beneficial in order to discover in exactly which cases stemming can be used with good results.

Acknowledgments

This work was partially supported by CNPq (Brazil) Projects 478979/2012-6, 309355/2011-7 and 305141/2015-5.

References

Alvares, R., Garcia, A., & Ferraz, I. (2005). STEMBR: A stemming algorithm for the Brazilian Portuguese language. *Progress in Artificial Intelligence – LNCS, 3808*, 693–701.

Baeza-Yates, R. A., & Ribeiro-Neto, B. (2011). *Modern information retrieval* (2nd ed.). Addison-Wesley Longman Publishing Co., Inc.

Billerbeck, B., Cannane, A., Chattaraj, A., Lester, N., Webber, W., Williams, H. E., & Zobel, J. (2004). RMIT University at TREC 2004. In *Proceedings of text retrieval conference (TREC).*

Brychcín, T., & Konopík, M. (2015). HPS: High precision stemmer. *Information Processing & Management, 51*(1), 68–91.

Buckley, C., & Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 33–40). ACM.

Dolamic, L., & Savoy, J. (2009). Indexing and stemming approaches for the Czech language. *Information Processing & Management, 45*(6), 714–720.

Fattah, M. A., Ren, F., & Kuroiwa, S. (2006). Stemming to improve translation lexicon creation form bitexts. *Information Processing & Management, 42*(4), 1003–1016.

Figuerola, C. G., Gómez, R., & de San Román, E. L. (2000). Stemming and n-grams in Spanish: An evaluation of their impact on information retrieval. *Journal of Information Science, 26*(6), 461–467.

Flores, F., Moreira, V., & Heuser, C. (2010). Assessing the impact of stemming accuracy on information retrieval. In T. Pardo, A. Branco, A. Klautau, R. Vieira, & V. de Lima (Eds.), *Computational processing of the portuguese language: Vol. 6001* (pp. 11–20). BerlinHeidelberg: Springer.

Harman, D. (1991). How effecttive is sufixing? *Journal of the American Society for Information Science, 42*(1), 7–15.

Hull, D. A. (1996). Stemming algorithms: a case study for detailed evaluation. *Journal of the American Society for Information Science, 47*(1), 70–84.

Jenkins, M.-C., & Smith, D. (2005). Conservative stemming for search and indexing: School of computing sciences.University of East-Anglia Norwich NR4 7TJ UK.

Jivani, A. (2011). A compaative study of stemming algorithms. *International Journal of Computer Technology and Applications, 2*(6), 1930–1938.

Jones, K. S., & Galliers, J. R. (1996). *Evaluating natural language processing systems: an analysis and review*. Springer-Verlag, New York, Inc.

Kraaij, W., & Pohlmann, R. (1995). Evaluation of a Dutch stemming algorithm. *The New Review of Document & Text Management, 1*, 25–43.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 191–202).

Levene, M. (2010). *An introduction to search engines and web navigation* (2nd ed.). Wiley.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics, 11*, 22–31.

Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P., & Datta, K. (2007). YASS: Yet another suffix stripper. *ACM Transactions on Information Systems, 25*(4), 18.

McNamee, P., Nicholas, C., & Mayfield, J. (2008). Don't have a stemmer? Be un+concern+ed. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 813–814). ACM.

McNamee, P., Nicholas, C., & Mayfield, J. (2009). Addressing morphological variation in alphabetic languages. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 75–82). ACM.

Méndez-Cruz, C.-F., Torres-Moreno, J.-M., Medina-Urrea, A., & Sierra, G. (2013). Extrinsic evaluation on automatic summarization tasks: Testing affixality measurements for statistical word stemming. In I. Batyrshin, & M. Mendoza (Eds.), *Advances in computational intelligence: Vol. 7630* (pp. 46–57). Berlin-Heidelberg: Springer.

Mollá, D., & Hutchinson, B. (2003). Intrinsic *versus* extrinsic evaluations of parsing systems. In *Proceedings of the EACL 2003 workshop on evaluation initiatives in natural language processing: Are evaluation methods, metrics and resources reusable?* (pp. 43–50). Association for Computational Linguistics.

Moral, C., Antonio, A. d., Imbert, R., & Ramirez, J. (2014). A survey of stemming algorithms in information retrieval. *Information Research: An International Electronic Journal, 19*(1).

Oard, D., Levow, G. A., & Cabezas, C. (2001). CLEF experiments at Maryland: Statistical stemming and backoff translation. *Cross-Language Information Retrieval and Evaluation, 2069*, 176–187.

Orengo, V. M., Buriol, L. S., & Coelho, A. R. (2007). A study on the use of stemming for monolingual ad-hoc portuguese information retrieval. In Proceedings of workshop of the cross-language evalution forum, *CLEF 2006 – LNCS: 4730* (pp. 91–98).

Orengo, V. M., & Huyck, C. R. (2001). A stemming algorithm for the Portuguese language. In *Proceedings of the 8th international symposium on string processing and information retrieval (SPIRE)* (pp. 183–193).

Paice, C. D. (1990). Another stemmer. *SIGIR Forum, 24*, 56–61.

Paice, C. D. (1994). An evaluation method for stemming algorithms. In *Proceedings of the 17th ACM SIGIR conference on research and development in information retrieval* (pp. 42–50).

Paice, C. D. (1996). Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science, 47*(8), 632–649.

Paik, J. H., Mitra, M., Parui, S. K., & Järvelin, K. (2011). GRAS: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems, 29*(4), 19.

Paik, J. H., Pal, D., & Parui, J. H. (2011). A novel corpus-based stemming algorithm using co-occurrence statistics. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, Beijing, China, July 24-28,* (pp. 863–872).

Pearson, K. (1895). *Note on regression and inheritance in the case of two parents: 58* (pp. 240–242).

Peng, F., Ahmed, N., Li, X., & Lu, Y. (2007). Context sensitive stemming for web search. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 639–646). ACM.

Peters, C., Braschler, M., & Clough, P. D. (2012). *Multilingual information retrieval – From research to practice*. Springer.

Peters, C., Clough, P. D., Gey, F. C., Karlgren, J., Magnini, B., Oard, D. W., & Stempfhuber, M. (2007). Evaluation of multilingual and multi-modal information retrieval. In *Proceedings of the 7th workshop of the cross-language evaluation forum, CLEF 2006: Vol. 4730*. Springer September 20-22, 2006, Revised Selected Papers.

Peters, C., Gey, F. C., Gonzalo, J., Müller, H., Jones, G. J. F., Kluck, M., & Rijke, M. d. (2006). Accessing multilingual information repositories. In C. Peters, F. C. Gey, J. Gonzalo, H. Müller, G. J. F. Jones, M. Kluck, B. Magnini, & M. d. Rijke (Eds.), *Proceedings of the 6th workshop of the cross-language evalution forum, CLEF 2005: Vol. 4022*. Springer 21–23 September, 2005, Revised Selected Papers.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Ramachandran, V. A., & Krishnamurthi, I. (2012). An iterative stemmer for Tamil language. In J.-S. Pan, S.-M. Chen, & N. Nguyen (Eds.), Intelligent information and database systems *(: Vol. 7198* (pp. 197–205). BerlinHeidelberg: Springer.

Saharia, N., Sharma, U., & Kalita, J. (2014). Stemming resource-poor Indian languages. *ACM Transactions on Asian Language Information Processing, 13*(3), 1–26.

Savoy, J. (1999). A stemming procedure and stopword list for general French corpora. *Journal of the American Society for Information Science, 50*(10), 944–952.

Savoy, J. (2006). Light stemming approaches for the French, Portuguese, German and Hungarian languages. In *Proceedings of the 2006 ACM symposium on applied computing* (pp. 1031–1035).

Sharma, D. (2012). Stemming algorithms: A comparative study and their analysis. *International Journal of Applied Information Systems, 4*(3), 7–12.

Sirsat, S., Chavan, V., & Mahale, H. (2013). Strength and accuracy analysis of affix removal stemming algorithms. *International Journal of Computer Science and Information Technologies, 4*(2), 265–269.

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management, 36*(5), 697–716.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on REsearch and development in information retrieval* (pp. 307–314). ACM.