Stemming in Spanish: A First Approach to its Impact on Information Retrieval

Carlos G. Figuerola, Raquel Gómez, Angel F. Zazo Rodríguez,
José Luis Alonso Berrocal
Universidad de Salamanca
Spain

Abstract

Most models and techniques employed in Information Retireval at some time or other use frecuency counts of the terms appearing in both documents and queries. Many words that derive from the same stem have a close semantic content. Locating stems common to several words and grouping them by replacing them with the corresponding stem can improve the working of these systems. Stemming procedures differ, however, depending on the different languages. We describe a stemmer for Spanish and the tests carried out by applying it to Information Retrieval.

1 Introduction

Most of the models and techniques employed in Information Retrieval use at some time or another frequency counts of the terms appearing in documents and queries. The concept of term in this context, however, is not exactly the same as that of word. Leaving to one side the matter of so-called empty words, which cannot be considered terms as such, we have the case of words derived from the same stem, which can be attributed a very close semantic content. [13]. The possible variations of the derivatives, together with their inflexions, alterations in gender and number, etc., make it advisable to group these variants under one term. If this is not done, a dispersion in the calculation of the frequency of such terms occurs and difficulty ensues in the comparison of queries and documents [21].

Moreover, the programs that are supposed to resolve the query must be able to identify the inflexions and derivatives -which may be different in the query and the documents- as similar and as corresponding to the same stem. Stemming, as a way of standardising the representation of the terms with which Information Retrieval systems operate, is an attempt to solve these problems.

However, the effectiveness of stemming has been the object of certain discussion, probably beginning with the work of Harman [9], who, after trying several algorithms (for English), concluded that none of them increased effectiveness in retrieval. Subsequent works [20] pointed out that stemming is effective as a function of the morphological complexity of the language being used, while Krovetz [17] found that stemming improves recall and even precision when documents and queries are short.

2 Previous Works

Stemming applied to Information Retrieval has been posed in several ways, from succinct stripping to the application of much more sophisticated algorithms. Study of it began in the 1960s with the aim of reducing the size of indices [3], and apart from being a way of standardising terms it can also be seen as a means to expand queries by adding inflexions or derivatives of the words to documents and queries.

Among the most well-known contributions we have the algorithm proposed by Lovin in 1968 [18], which is in some sense the basis of subsequent algorithms and proposals, such as those of Dawson [5], Porter [21] and

Paice [19]. Although a good part of the works are oriented to use with documents in English, it is possible to find proposals and algorithms for specific languages, among them Latin, in spite of its being a dead language [30], Malaysian [2], French [28], [29], Arabic [1], Dutch [15], [16], Slovene [20] and Greek [14].

As regards Spanish, diverse stemming mechanisms were applied to Information Retrieval operations in some of the TREC conferences (Text Retrieval Conference) [12]. In general, these applications consisted in using the same algorithms as for English, but with suffixes and rules for Spanish. Regardless of the algorithms applied, and of their adaptation to Spanish, the linguistic knowledge used (lists of suffixes, rules of application, etc., was quite poor [6].

From the perspective of language processing, in recent years several stemmers and morphological analysers for Spanish have been developed, among which we have the COES [23] tools, made available to the public by its authors at http://www.datsi.fi.upm.es/ coes/ under GNU licensing; the morpho-syntax analyser MACO+ [4] (http://nipadio.lsi.upc.es/cgi-bin/demo/demo.pl) or the FLANOM / FLAVER stemmers [27], [26] (http://protos.dis.ulpgc.es/). However, we are unaware of any experimental results of their application to Information Retrieval.

On the other hand, on several occasions the use of n-grams has been proposed to obviate the problem posed by inflexions and derivatives of words [22]. In prior works, however, we were able to verify the scant effectiveness of this mechanism from the point of view of Information Retrieval [8], as well as the inadequacy of the well-known Porter algorithm for languages such as Spanish.

3 The Stemmer

The basis of the stemmer consists of a finite states machine that attempts to represent the modifications undergone by a stem when a certain suffix is attached or added to it. There is thus an instance of this automaton for each suffix contemplated: each of these implies a series of rules expressing how that suffix is incorporated into the stem. Since, for one same suffix, there may be a large number of variants and exceptions, on occasion the resulting automaton can be quite complex.

Thus, in order to stem a word, the longest suffix coinciding with the end of this word is sought and the corresponding automaton is formed with the rules for that suffix. The network of this automaton is searched with the word to be stemmed and the chains obtained in the terminal node of the automaton are contrasted with a dictionary of stems. If the chain obtained is found in the dictionary the stem is considered to be correct.

Taking into account that the transformations may occasionally overlap, adding more than one suffix, the process is repeated recursively until the correct stem is found. If, once the possibilities are exhausted, none of the terminal chains obtained are found in the dictionary of stems, it is deduced that either the word can be considered as standardised in itself, or else it is a case not foreseen by the stemmer.

This last instance may mean the following

- a) the word has a suffix that is not on the list of suffixes of the stemmer
- b) the suffix is added in a way unforeseen by the rules incorporated into the knowledge base
- c) the stem is not in the dictionary of stems.

This allows the stemmer to be subjected to a training process in which the results of stemming the words of a corpus are examined manually and the knowledge base of the stemmer is corrected when necessary

Now, we can distinguish between two classes of stemming: flexional a nd derivative. Whereas the former has clear and defined limits, this does not occur with the latter. Moreover, the semantic distance between two different flexions of the same stem can in general be considered of little importance (for example, *libro* and *libros*), whereas the semantic difference between a stem and its derivatives may be great; for example, *sombra* (shade), *sombrilla* (parasol, sunshade), and even *sombrero* (hat).

As for flexional stemming, it should heed changes in gender and/or number for nouns and adjectives and changes in person, number, tense and mode for verbs. Treatment of nouns and adjectives is simple, since both the change in gender and number follow simple rules; exceptions to the rules exist but there are few and they an be treated individually. Verbs, however, are another case. Besides the great number of forms a verb can take, the

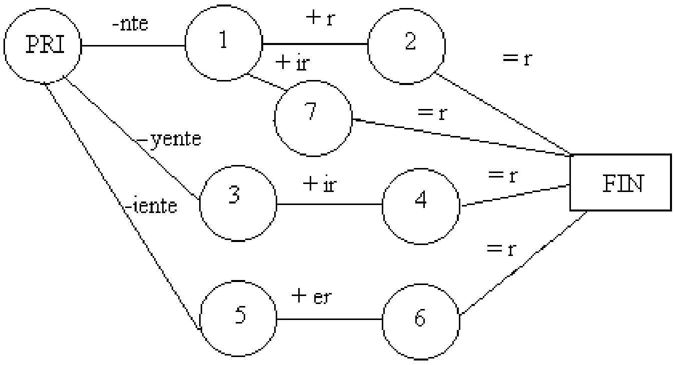

Figure 1: Automaton for the suffix *-ente*

main problem lies in the large amount of irregular verbs in Spanish. There may be more than 40,000 irregular verbs s and any basic course of Spanish includes lists of 8 or 10 thousand irregular verbs. Fortunately, these can be grouped into approximately 80 different models, although they do not always strictly follow a specific model and there are many exceptions.

In flexional stemming there is another complex problem to be solved: the grammatical ambiguity of many words. A certain word ending in a certain suffix may pertain to different grammatical categories and, depending on which it pertains to, the flexional transformations it has undergone will be different and will in consequence have come from different stems. A simple example would be the word *colecciones*: it could be the plural of the noun *colección* (collection) or else the second person singular present subjunctive of *coleccionar* (to collect), and would thus give rise to two different stems.

The way to solve this ambiguity could lie in resorting to the specific context of the word and determining its grammatical category, to then choose the right stem. Our stemmer cannot yet resolve this ambiguity. However, one should take into account that some forms are more frequent than others; a verb in subjunctive mode is much more infrequent than a noun, and even more so in journalistic texts such as the ones we have dealt with.

For the moment, until we manage to solve this ambiguity, our stemmer chooses the most frequent stems; this necessarily introduces an element of error, but since it always applies the same stem that error is always less than it would be without stemming. Furthermore, derivation produces a much higher number of forms based on one stem. Flexional transformations can occur on any of these forms and therefore derivative stemming should be carried out after flexional stemming; for example, *libreros* (book-sellers) is a plural noun that should be reduced to singular in order to eliminate the suffix and end up with the stem *libro* (book).

4 The impact of stemming on Information Retrieval

The 40 queries of the CLEF spanish monolingual collection were executed in three modalities: without stemming, applying flexional stemming and applying flexional plus derivational stemming. Obviously, the stemming was applied both to documents and queries, and in all three cases empty words were eliminated previously, based on a standard list of 538 (articles, conjunctions, prepositions, etc.).

The algorithm is the same for both flexional and derivative stemming. What changes, obviously, are the suffixes and rules of application, as well as the dictionary or list of stems to be used. For flexional stemming the number of suffixes considered was 88, with a total of 2,700 rules of application. The dictionary of stems consists of 80,000 entries. For derivative stemming the number of suffixes is higher (since i t is actually a matter of flexion plus derivatives): 230 with 3,692 rules of application. The dictionary or list of stems is much shorter: approximately 15,000 stems.

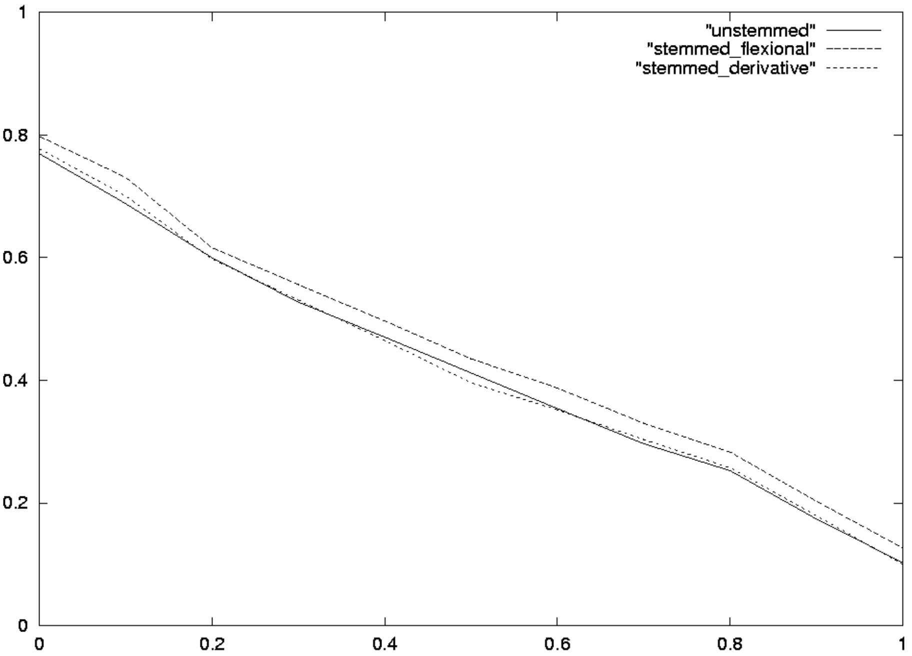

Figure 2: Results of the official runs

After eliminating empty words, the document collection produced a total of 36,573,577 words, with 353,868 unique words. Flexional stemming reduced these 353,868 unique words to 284,645 stems; nevertheless, of these, 141,539 (almost half) were stems that appeared only once in the document collection. A simple glance shows that a good part of them correspond to typographical errors (which cannot be stemmed without previous detection and correction), as well as to proper names, acronyms, etc. Derivative stemming reduced the number of stems: the 353,868 unique words produced 252,494 single stems. Of these, 127,739 appeared only once in the document collection; most of them are typographical errors.

4.1 The Retrieval Model

To execute or solve the queries we used our own retrieval engine, Karpanta, [7], which is based on the well known vectorial model, defined by Salton some time ago [25]. The weights of the terms were calculated according to the usual scheme of *Frequency of term in the document x IDF*. IDF (Inverse Document Frequency) is an inverse function of the frequency of a term in the entire collection (understood as the number of documents in which it appears) [10]. The similarity between each document and each query was calculated using the formula of the cosine, as is usual in these cases [24].

Taking into account that our objective was to evaluate the effect of stemming, we did not consider it necessary to apply additional techniques such as feedback of queries [11], although the Karpanta retrieval system permits it. Actually, our intention was not so much to achieve the best results, but to measure the differences among the results obtained with each of the three modalities mentioned above.

5 Results, (non) Conclusions and Future Work

The results can be seen in the attached plot, and they are some disappointing. The differences among the cases are scarce. Flexional stemming produces only about 3 % of improvement over unstemming. Derivative stemming is even a litle bit worse than no stemming.

Journalistic texts are specially plain in morphology and syntax, but we don't know if that can to explain the small difference between stemmed and unstemmed runs. The low time between the release of results and the dead date for these worknotes don't let us study in depth about the causes of these results.

For the future, we must to finish the stemmer, specially resolving the ambiguity between words which can have diferents stems. This can be achieved by means of the context in which the word occures. In adition, it has already been noted that great semantic differences can exist between a stem and its derivatives. In this sense, it is worth asking whether a detailed study of derivative suffixes and a selective application of stemming could avoid this problem, i.e. whether there are suffixes that produce derivatives semantically very distant from the original stem and vice-versa. Another, non-exclusive, possibility is the one noted by Krovetz [17] of using thesauri (or other methods) to determine the semantic relation between stem and derivative.

References

[1] H. Abu-Salem, M. Al-Omari, and M. W. Evens. Stemming methodologies over individual queries words for an arabian information retrieval system. *JASIS*, 50(6):524–529, 1999.

[2] F. Ahmad, M. Yussof, and M. T. Sembok. Experiments with a stemming algorithm for malay words. *JASIS*, 47(12):909–918, 1996.

[3] C. Bell and K. P. Jones. Toward everyday languaje information retrieval system via minicomputer. *JASIS*, 30:334–338, 1979.

[4] J. Carmona, S. Cervell, L. Márquez, M. Martí, L. Padrón, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo. An environment for morphosyntactic processing of unrestricted spanish text. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain, 1998.

[5] J. Dawson. Suffix removal and word conflation. *ALLC bulletin*, 2(3):33–46, 1974.

[6] C. G. Figuerola. La investigación sobre recuparación de la información en español. In V. Gonzalo García, C. y García Yebra, editor, *Documentación, Terminología y Traducción*, pages 73–82, Madrid, 2000. Síntesis.

[7] C. G. Figuerola, J. L. Alonso Berrocal, and A. F. Zazo Rodríguez. Disseny d'un motor de recuperació d'informació per a ús experimental i educatiú = diseño de un motor de recuperación de información para uso experimental y educativo. *BiD. textos universitaris de biblioteconomia i documentació*, 4, 2000.

[8] C. G. Figuerola, R. Gómez, and E. López de San Román. Stemming and n-grams in spanish: an evaluation of their impact on information retrieval. *Journal of Information Science*, 26(6):461–467, 2000.

[9] D. Harman. How effective is suffixing? *JASIS*, 42(1):7–15, 1991.

[10] D. Harman. Ranking algorithms. In *Information retrieval: data structures and algorithms*, pages 363–392, Upple Saddle River, NJ, 1992. Prentice-Hall.

[11] D. Harman. *Relevance Feedback and Others Query Modification Techniques*. Prentice-Hall, Upple Saddle River, NJ, 1992.

[12] D. Harman. The trec conferences. In *Proceedings of the HIM'95 (Hypertext-Information Retrieval-Multimedia)*, pages 9–23, 1995.

[13] D. HULL. Stemming algorithms: a case study for detailed evaluation. *JASIS*, 47(1), 1996.

[14] T. Z. Kalamboukis. Suffix stripping with moderm greek. *Program*, 29(3):313–321, 1995.

[15] W. Kraaij and R. Pohlmann. Porter's stemming algorithm for dutch. In L. G. M. Noordman and W. A. M. de Vroomen, editors, *Informatiewetenschap*, Tilburg, 1994. STINFON.

[16] W. Kraaij and R. Pohlmann. Viewing stemming as recall enhancement. In *SIGIR 96*, pages 40–48, 1996.

[17] R. Krovetz. Viewing morphology as an inference process. In *SIGIR 93*, pages 191–203, 1993.

[18] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[19] C. D. Paice. Another stemmer. In *SIGIR 90*, pages 56–61, 1990.

[20] M. Popovic and P. Willet. The effectiveness of stemming for natural-language access to slovene textual data. *JASIS*, 43:384–390, 1992.

[21] M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[22] A. Robertson and P. Willet. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):28–47, 1999.

[23] S. Rodríguez and J. Carretero. A formal approach to spanish morphology: the coes tools. In *XII Congreso de la SEPLN*, pages 118–126, Sevilla, 1996.

[24] G. Salton. *Automatic Text Processing*. Adisson-Wesley, Reading, MA, 1989.

[25] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[26] O. Santana, J. Pérez, F. Carreras, J. Duque, Z. Hernández, and G. Rodríguez. Flanom: Flexionador y lematizador automático de formas nominales. *Lingüística Española Actual*, XXI(2):253–297, 1999.

[27] O. Santana, J. Pérez, Z. Hernández, F. Carreras, and G. Rodríguez. Flaver: Flexionador y lematizador automático de formas verbales. *Lingüística Española Actual*, XIX(2):229–282, 1997.

[28] J. Savoy. Effectiveness of information retrieval systems used in a hypertext environment. *Hypermedia*, 5:23–46, 1993.

[29] J. Savoy. A stemming procedure and stopword list for general french corpora. *JASIS*, 50(10):944–952, 1999.

[30] R. Schinke, A. Robertson, P. Willet, and M. Greengrass. A stemming algorithm for latin text databases. *Journal of Documentation*, 52(2):172–187, 1996.