



An evaluation of conflation accuracy using finite-state transducers

Carmen Galvez and Félix de Moya-Anegón

Department of Information Science, University of Granada, Granada, Spain

328

Received February 2005
Revised July 2005
Accepted July 2005

Abstract

Purpose – To evaluate the accuracy of conflation methods based on finite-state transducers (FSTs).

Design/methodology/approach – Incorrectly lemmatized and stemmed forms may lead to the retrieval of inappropriate documents. Experimental studies to date have focused on retrieval performance, but very few on conflation performance. The process of normalization we used involved a linguistic toolbox that allowed us to construct, through graphic interfaces, electronic dictionaries represented internally by FSTs. The lexical resources developed were applied to a Spanish test corpus for merging term variants in canonical lemmatized forms. Conflation performance was evaluated in terms of an adaptation of recall and precision measures, based on accuracy and coverage, not actual retrieval. The results were compared with those obtained using a Spanish version of the Porter algorithm.

Findings – The conclusion is that the main strength of lemmatization is its accuracy, whereas its main limitation is the underanalysis of variant forms.

Originality/value – The report outlines the potential of transducers in their application to normalization processes.

Keywords Linguistics, Semantics, Programming and algorithm theory, Accuracy

Paper type Research paper

Introduction

Conflation is the process of matching and grouping together variants of the same term that are semantically equivalent. A variant is defined as a text occurrence that is conceptually related to an original term and can be used to search for information in text databases (Sparck Jones and Tait, 1984; Tzoukermann *et al.*, 1997; Jacquemin and Tzoukermann, 1999). This is done by means of computational procedures known as conflation algorithms, whose primary goal is the normalization of uniterms and multiterms (Galvez *et al.*, 2005). Uniterm conflation algorithms take into account the common endings of the words that can be conflated. The programs that carry out this process are called: stemmers, when the process involves non-linguistic techniques and stemming algorithms, and lemmatizers, when linguistic techniques and lemmatization algorithms are used.

A stemmer tries to reduce various forms of a word to a single stem, defined as the “base form,” from which inflected forms are derived. A common method of stemming is affix removal based on a list of affixes and rules. A stemmer, however, operates on a single word without knowledge of the context, and therefore cannot discriminate words that may have different meanings depending on the context of their appearance. At the same time, stemmers are typically easy to implement, and run fast, yet they do not give a high percentage of accuracy, making them inappropriate for some applications.



A lemmatizer attempts to obtain the lemma, defined as the combination of the stem and its part-of-speech (POS) tag, which defines the role of terms in a sentence. The correct identification of the syntactical category of a word in a sentence requires knowledge of the grammar of a language, implying natural language processing (NLP). A well-known method of lemmatization consists of a morphological analysis of the variants and their reduction to lemmas. The process of lemmatization with finite-state technology consists of standardizing the terms according to a dictionary look-up, or a lexicon, that is configured as a lexical database used to treat as equivalent forms certain entry terms, related with the canonical form or lemma.

Literature review and evaluation measures

Our general understanding is that literature regarding automatic conflation methods in IR can depart from one of several frameworks that are not mutually exclusive, taking the following criteria into account:

- non-linguistic vs linguistic techniques;
- language independent vs dependent techniques; and
- similarity vs equivalence relations.

Within this general structure, we may classify the different means of reducing morphological variants in IR as elimination of affixes, stemming, word segmentation, *n*-grams, and linguistic morphology (Lennon *et al.*, 1981). A categorization of methods for reducing morphological variants begins with the distinction between manual methods and automatic ones, the latter including: affix removal, successor variety, *n*-gram matching, and table lookup (Frakes, 1992), whereas the conflation techniques employed habitually in IR are stemming and lexical lookup (Paice, 1996).

Conflation based on stemming techniques involves the elimination of the longest possible affixes, and so the algorithms applied in this way are known as longest match or simple-removal algorithms. The ones most often used with the English language are those of Lovins (1968), Dawson (1974), Porter (1980) and Paice (1990). The Porter algorithm, available at the Snowball web site (2003), has been implemented with French, Spanish, Italian and Portuguese, as well as with German, Norwegian, Swedish, and other languages.

The best known string-similarity algorithms are those based on *n*-gram similarities, the *n*-gram of a string being any substring of some fixed length. These have been extensively applied to IR-related tasks such as query expansion (Adamson and Boreham, 1974; Lennon *et al.*, 1981; Cavnar, 1994; Damashek, 1995; Robertson and Willett, 1998). They are used as well for automatic spelling correction (Angell *et al.*, 1983; Kosinov, 2001), based on the assumption that the problems of morphological variants and spelling variants are similar.

In language-dependent linguistic techniques, dictionaries are utilized to fuse lexical variants into lemmas, by means of lemmatization algorithms. The first computational implementation of this approach was with the PC-KIMMO parser (Karttunen, 1983), later used as the scheme for the Xerox morphological analyzer developed by the Multi-lingual Theory and Technology Group. One of the top applications of the Xerox tool, designed for morphological parsing using finite-state technology, is the reduction of lexical variants in IR systems. The XEROX-XRCE analyzer has been applied to English, Dutch, German, Hungarian, French, Italian, Portuguese, and Spanish.

More recent developments involve Czech, Danish, Finnish, Norwegian, Polish, Romanian, Russian, Swedish and Turkish. A further tool based on finite methods is the English morphological analyzer ENGTWOL (Voutilainen, 1995). Lemmatizers developed for Spanish include the COES tool (Rodríguez and Carretero, 1996), and the morphological analyzer MACO (Carmona *et al.*, 1998).

Evaluation parameters of conflation methods

The evaluation of term conflation methods can be found upon three essentially different measures:

- (1) *Evaluating IR effectiveness.* In terms of standard external measures – of outcome on retrieval performance. Numerous studies compare the effectiveness of automatic conflation by determining the retrieval performance with test collections (Lennon *et al.*, 1981; Harman, 1991; Hull, 1996). Results to date have been diverse and certainly not always positive (Frakes, 1992). Whereas for languages with a simple morphology like English retrieval is not particularly successful, an experimental result found that stemming improved recall and precision when documents and queries are short (Krovetz, 1993). Other works show that the language of the document involved is an important factor; experiments show stemming is beneficial for highly inflected languages (Popovic and Willett, 1992; Savoy, 1993; Kraaij and Pohlmann, 1994, 1995; Pirkola, 2001).
- (2) *Evaluating the index compression factor (ICF).* In terms of fractional reduction measures in index size – of effects on compression performance. In order to reduce index space for inverted files, stemming and lemmatization can have a marked effect on compression performance. The ICF is defined as the reduction in index size acquired by means of conflation methods and can be calculated by the equation: $ICF = (N - S)/N$, where N is the number of unique words in the corpus before stemming/lemmatization, and S is the number of unique stems/lemmas after stemming/lemmatization (Frakes and Fox, 2003). Research by Lennon *et al.* (1981) showed compression percentages for various stemmers and databases (Cranfield, National Physical Laboratory, INSPEC and Brown Corpus), sometimes reducing the size of files by as much as 50 percent. Although the issues related to storage are not a problem now, a small number of reports have looked into term conflation as a method for index compression.
- (3) *Evaluating accuracy.* In terms of internal measures to determine the correctness – of effects on conflation performance. Here, the concept of “correct lemma/stem” will not refer to linguistic correctness, but to the capacity of the lemmatizer or stemmer to actually merge term variants into a single lemma or stem. Because conflation procedures are prone to error, diverse studies have been carried out to identify the sources of error. In stemming procedures, the inaccuracies appear in the form of understemming errors, which occur when words that refer to the same variants are not reduced to the same stem; and overstemming errors, which occur when words are stemmed incorrectly because they are not actual variants (Xu and Croft, 1998). An assessment approach for stemming algorithms was developed by Paice (1996), evaluating accuracy of a stemmer by counting the actual understemming and

overstemming errors it commits. His measure provides insights which might help in stemmer and lemmatizer optimization. Among the works under this approach we also find those carried out by Tzoukermann *et al.* (1997) and Nakov (2003).

Most evaluations of term conflation methods have focused on retrieval performance, yet there is lack of research surrounding conflation accuracy. In spite of the importance of the works on conflation effectiveness in IR, these experiments usually provide no information about the type of errors committed. Though linguistic correctness is regarded as irrelevant in IR (Savoy, 1993), the errors and inaccurately stemmed/lemmatized forms would lead to retrieval of inappropriate documents, besides producing a negative effect in applications such as text classifications and the preprocessing stages of text application subproblems, including text mining or information extraction (IE).

Related work on Spanish – participations in TREC and CLEF

In Spanish, conflation methods are needed to standardize term variants because of the abundance of morphological and lexical phenomena. Most work involving the Spanish language has focused on evaluating the effectiveness of IR. Along these lines, diverse conflation algorithms were presented in Text REtrieval Conferences (TREC):

- In TREC 3, a team from the University of Cornell presented an application of the SMART information retrieval system for Spanish (Buckley *et al.*, 1994), asserting that only elementary stemming is needed, and that a stop-word list can be developed from the most frequent stems (manually reviewed to choose which stems should be kept). In contrast, the Center for Intelligent Information Retrieval (CIIR), at the University of Massachusetts, presented an application of the INQUERY retrieval system (Broglio *et al.*, 1994) arriving at the conclusion that sophisticated stemming produces a significant improvement for Spanish.
- In TREC 4, it was demonstrated that the results with the application of the SMART system are not dependent on the language used (Buckley *et al.*, 1995); meanwhile, experiments with the INQUERY system (Allan *et al.*, 1995) underlined the ambiguity of Spanish terms. Regardless of the finding from Cornell and Massachusetts that it might not improve performance, further work from the University of Berkeley (Gey *et al.*, 1995) strove to develop a Spanish stemmer, applying two approaches – one which attempts to group verb variants into a standardized form, the other involving a massive stop-word list of variants of common words. The conclusion was that stemming works as well for Spanish as it does for English. On the other hand, experiments developed by the Xerox Research Center (Hearst *et al.*, 1995) used a finite-state lemmatizer and hidden Markov model based on POS tagging to conflate Spanish language text into canonical forms in the context of IR, and found its performance to be consistently better.
- In TREC 5, a research group from Dublin City University presented an evaluation of the performance of the Porter stemming algorithm for Spanish, employing query space reduction techniques (Kelledy and Smeaton, 1996); the experiments reported improvements in retrieval efficiency through query space thresholding. Other participants merely tried to improve the algorithms utilized

in previous conferences (Allan *et al.*, 1996; Buckley *et al.*, 1996), while Xerox tested its lemmatizer in English/Spanish cross-language retrieval using English versions of the query and an English-Spanish bilingual implementation, the latter resulting slightly more effective (Hull *et al.*, 1996).

Overall, the TREC participants showed that results of conflation methods for Spanish are similar, or just slightly better, than for English. In a different environment of evaluation, the tests on Spanish under the European Cross-language Evaluation Forum (CLEF) arrived at similar results. In CLEF 2001, work entailing a stemmer based on finite-state automata for Spanish improved effectiveness by only 3 percent for inflectional stemming with respect to understemming (Figuerola *et al.*, 2002). Later, the COLE Group in CLEF 2002 (Vilares *et al.*, 2003) applied NLP techniques in IR, in particular a tagger-lemmatizer developed by Graña *et al.* (2001), and compared the results obtained with Porter's stemmer used by the open source search engine Muscat[1]; the conclusions were that lemmatization performs better than stemming.

Objectives of the study

The position taken in this paper is to evaluate the conflation procedure not by its effect on query success for IR purposes, but through direct consideration of the set of terms that are correctly conflated. In this study, we investigate a lemmatization process, and compare it with other stemming processes, measuring their ability to conflate terms from a test corpus. Our objectives can be summed up as:

- to defend finite-state technology for the unification of term variants into canonical lemmatized forms;
- to implement a linguistic development environment based on the technology of finite-state transducers (FSTs) for Spanish that allows for term conflation; and
- to evaluate the accuracy of term conflation, comparing the results with those obtained under a stemmer based on Porter's algorithm.

We shall now briefly describe the computational and linguistic base of the lexical analysis of our approach and present a careful survey, with emphasis on the influence of the formal language theory on lexical analysis, that will help us determine the origin of problems and offer some explanation of the grounds for accuracy of term conflation using FSTs. The tools developed with this technology are only capable of analyzing and conflating those terms previously stored in the lexicon, and irregularities of the lexical inflections can interfere with establishing proper equivalency between the surface and the lexical forms stored.

The term conflation process via finite-state technology

Formal language theory focuses on languages that can be described in very precise terms, such as programming languages. Natural languages are not formal, as no well-defined boundary exists between correct sentences or those that are incorrect. Notwithstanding, formal definitions approximating natural language phenomena can be encoded into computer programs and be used for the automated processing of natural language. Likewise, formal descriptions can be utilized by linguists to express theories about specific aspects of natural languages, including morphological analysis.

The most important application of the formal language theory to linguistics came from Chomsky (1957): his basic hypothesis was that the different types of formal languages were capable of modeling natural language syntax. This theoretical foundation beneath formal languages and grammars has a direct relation with the theory of machines or automata, abstract devices able to receive and transmit information. Though Chomsky demonstrated that natural language syntax cannot be modeled via finite-state devices, there are still many subsets of natural language that can be correctly described by very simple means, as occurs with the rules of phonology and morphology.

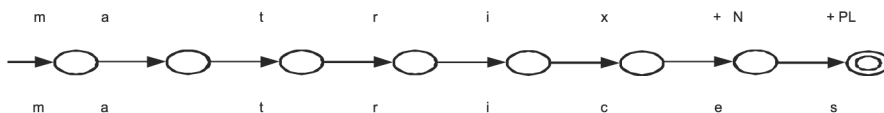
Johnson (1972) was the first to observe that phonological rewrite rules could be represented by finite-state devices. His “two-level model” formalism is founded on the generative phonology of Chomsky and Halle (1968). The formal properties of two-level rules differ from generative rules in the following essential ways, as stated by Antworth (1995):

- they define a correspondence between the underlying and surface symbol (as opposed to transformation);
- they are applied simultaneously (not sequentially); and
- they are bidirectional (not unidirectional).

The insight of the two-level model was key in progressing from a computational model for word-form recognition and generation, as evidenced in the two-level morphology of Koskenniemi (1983). Because morphology studies the “rules of word formation,” the Koskenniemi model further establishes a correspondence between the lexical form, or canonical form, and the surface form of the words, to be represented using finite-state technology.

The underlying assumption of the finite-state approach to morphology is that the relation between surface forms and the corresponding lexical forms can be described as a regular or rational relation $R(T)$, defined using the metalanguage of regular expressions (Karttunen *et al.*, 1992). With a suitable compiler, the $R(T)$ source code can be compiled into a FST that implements the relation computationally. The two-level analyzers match surface forms, or variants, with lexical forms, or lemmas, and vice versa. This also means that they can be used for either the recognition of strings or for the generation of strings. The lexical forms are configured as regular expressions, in this case lexical expressions stored as lists of canonical forms, plus a set of morphological rules, applied in a parallel manner between the two levels, and compiled in an FST (Figure 1).

In the two-level model developed by Koskenniemi (1983), the morphological processing entails two basic areas:



Source: Adapted from Karttunen *et al.* (1992)

Figure 1.
Transducer path

- (1) morphotactics, or the study of the formation of words according to the combination of morphemes, or smaller units of meaning; and
- (2) morphological alternations, or the study of modifications depending on the context of appearance.

The two-level rules represent valid combinations of morphemes and take charge of mapping surface strings onto lexical strings. Each rule is linked to a transducer that codifies some limitation on the equivalence (Figure 2). For instance, for the correspondence between the surface form “matrices” and the lexical form “matrix,” one rule compiled in an FST has the function of transforming $c \rightarrow x$ in a specific phonological context, and another rule says to eliminate an e when it appears after ac and before an s .

At first glance, Koskenniemi’s model would appear to have a simple, direct application through finite-state technology. Yet when we stop to consider the great variety of spelling rules, and the morphological alternations produced by the stringing together of morphemes, it becomes obvious that the correspondence between surface strings and lexical strings will not always be attained under this straightforward model.

The problem of mapping variant forms to canonical forms

To solve the problem of mapping variant forms to canonical forms, a mathematical property is adopted: regular relations are closed under composition (Kaplan and Kay, 1981). That is, if we have two rules that are sequentially applied through two transducers that feed each other, so that the output of the first transducer is the input of the second, a new equivalent transducer can be designed by means of an operation of composition. This shows that between the underlying or lexical level and the surface level, there are intermediate levels comprising a system of rules applied in a sequential manner, which can be depicted as an FST cascade, or series of vertically connected transducers. The intermediate levels and the symbols active in them can be eliminated by combining the different transducers to obtain a new one that would operate only on two-levels (lexical and surface).

Departing from a special interpretation of Koskenniemi’s (1983) model, the lexical transducer developed by Karttunen *et al.* (1992) works with two properties of calculus applied through finite-state technology: intersection and composition of FSTs. Basically, this new lexical transducer has two components:

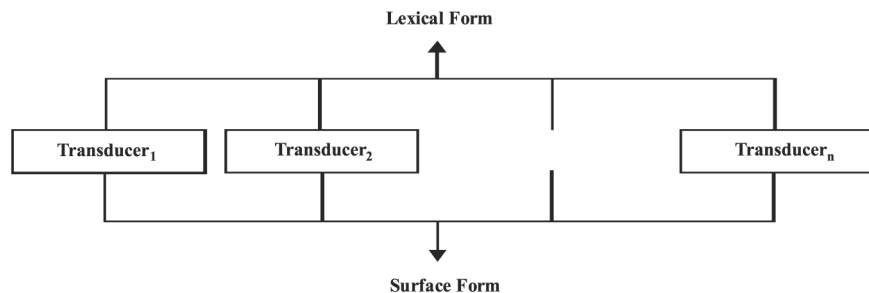


Figure 2.
Construction of an FST in parallel

Source: Adapted from Koskenniemi (1983)

- (1) The lexical component – lexicon or dictionary – compiled in an FST, which defines the set of valid canonical or lexical forms of the language, where the POS tags are part of canonical forms.
- (2) The rule component, or set of rules compiled in an FST, which is in charge of assigning lexical forms to all the surface occurrences, and vice versa. Therefore, the rules establish the conditions for which the information deposited in the dictionary of canonical forms is applied successfully and is joined to the lexicon by means of the aforementioned mathematical operations of intersection and composition.

The operations of intersection and composition are used to process the regular alternations. The greater the number of rules, the more extensive the computational resources required to process irregularities (Karttunen, 1994). The formalism we propose can describe regular and irregular variations without FST cascades or two-level rules: the irregularities are represented directly in graphic transducers. Essentially, three analytical resources are needed – a dictionary of canonical forms, a dictionary of inflectional forms, and a dictionary of frozen expressions and compound lemmas – to recognize and group variants, and all three are constructed with the help of a linguistic toolbox (Silberztein, 1993, 2000). Large-coverage dictionaries have been built using this methodology for English, French, Spanish, German, Greek, Italian, or Russian, among other languages.

Research design

A dataset obtained from a specialized database in Spanish was used to design the tools of lexical analysis, and then later used as the corpus to evaluate our results. The data-oriented approach made it possible for us to adapt the system in order to deal with expressions specific to the domain of information science that could not be processed with general-purpose analyzers. Exhaustive lexicons are costly, if not impossible, to produce; and the effort invested in built-in large-coverage dictionaries would have been enormous, exceeding the practical objectives of this study. Nevertheless, this mode of procedure should not entail bias, as it means that we can avoid words that cannot be conflated because they are not in the dictionary.

Constructions of electronic dictionaries

The construction of lexical resources was founded on the premise that between inflectional forms and canonical forms there exists an equivalence relation, which can be represented by means of FSTs. Our approach involves using a graphic interface, *FSTGraph*, which enabled us to draw FSTs (Silberztein, 1993, 2000). This linguistic tool also aided in the construction of handcrafted electronic dictionaries with which to reduce the terms of a corpus to lemmas. We must point out as well that we do not deal with compound terms or multiterms, which should be analyzed together. In our study, multiterms were broken down and lemmatized separately.

The first matter at hand is to determine the information to be stored in the dictionary. The second matter is to decide on which grammatical and morphological distinctions are relevant for the processing and the recognition of the lexical units. In many cases, the lexicon is developed *ad hoc* for a specific application, and it usually takes on one of two forms:

- (1) Exhaustive lists of all the lexical entries of the language to be analyzed. A simple list of all the lexical forms that can appear in a language would result into vast a construction, and some type of limitation is necessary.
- (2) Partial lists of the lexical entries to distinguish, on the one hand, the lemmas or stems of the words, and on the other hand, how the affixes are joined to the stems through the morphological processes of inflection and derivation.

The application that we adopt for representation of inflectional information consists of partial lists, in this case electronic dictionaries, represented internally by FSTs. These electronic dictionaries contain canonical forms with syntactic codes to indicate the POS category, and each code is linked to a graphic FST made up of an initial node and a final node that describe the path the morphological analyzer should trace. In order to produce the inflected forms, characters are deleted from the lemma using a delete character operator (*L*), which does not require two-level rules or finite-state calculus (Silberztein, 1993, 2000). For instance, all the inflected forms of given nouns associated with the same inflectional paradigm can be automatically generated through the association of lemmas to an inflectional code, as shown in Figure 3.

Transducers are projected upon canonical forms, automatically producing electronic dictionaries with inflected forms that contain the canonical forms along with the inflected forms, the POS categories, and inflectional information. The variations are represented directly in a graph editor. FST associates sets of suffixes with the corresponding inflectional information that would affect a large class of similar lexical items.

All operations are performed by means of a graphic interface that allows us to draw inflection transducers. Once the FSTs are compiled, they are projected upon the dictionary of canonical forms (known as DELAS), automatically producing the expanded dictionary of inflected forms (DELAF), along with canonical forms, POS categories, and inflectional information (Table I). A DELACF dictionary could contain, in addition, frozen expressions, acronyms or abbreviations.

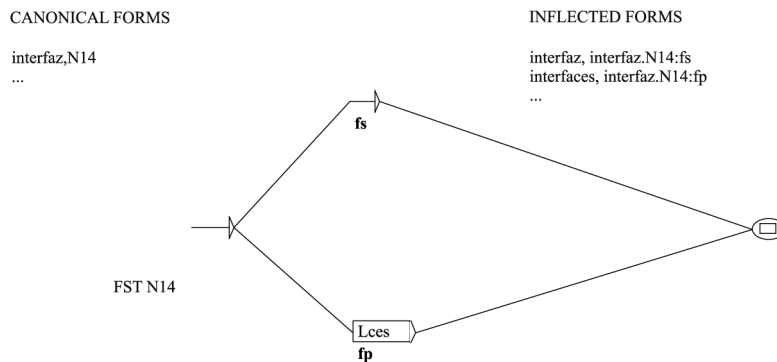


Figure 3.
The FST N14 associates certain lemmas, belonging to a inflectional paradigm

Notes: The inflectional paradigm has the corresponding inflectional codes (*f*’ feminine, *s*’ singular and *p*’ plural), in order to obtain the inflected forms from the lemmas (the final letter ‘z’ of the lemma should be eliminated using the delete operator *L*, *Left*)

Spanish DELAS dictionary	Spanish DELAF dictionary
a, PREP	a, a.PREP
abajo, ADV	abajo, abajo.ADV
abatir, V3	abata, abatir.V3:S1s:S3s
abierto, A1	abatamos, abatir.V3:S1p
abogar, V103	abatan, abatir.V3:S3p
abordar, V1	abatas, abatir.V3:S2s
abreviado, PA	abate, abatir.V3:P3s:Y2s
abreviar, V1	abaten, abatir.V3:P3p
abreviatura, N5	abates, abatir.V3:P2s
abrir, V3	abatid, abatir.V3:Y2p
absolutamente, ADV	abatido, abatir.V3:P
abstraer, V202	abatiendo, abatir.V3:G
Absys, N + PR	abatiera, abatir.V3:IS1s:IS3s
academia, N5	abatierais, abatir.V3:IS2p
acaparar, V1	abatieran, abatir.V3:IS3p
acceder, V2	abatieras, abatir.V3:IS2s
accesible, A6	abatieron, abatir.V3:J3p
acceso, N4	abatiese, abatir.V3:IS1s:IS3s
aceptar, V1	abatieseis, abatir.V3:IS2p

Table I.
Entries of the electronic
dictionaries

The entries of the main dictionaries contain the following elements:

- (1) Dictionary of lemmas, with:
 - canonical forms, or lemmas, selected from the binary opposition of the unmarked terms, or negative terms, and the marked or positive terms. For instance, within the general category N (noun) and A (adjective) we select the unmarked terms, which are masculine/singular, and in the category V (verb) we select infinitive; and
 - POS tags, represented by the following codes: N (noun), V (verb), A (adjective), ADV (adverb),...
- (2) Expanded dictionary of inflected forms (DELAF). The entries of this dictionary are generated automatically from the dictionary of lemmas, and contain:
 - inflected form;
 - lemma;
 - POS tag; and
 - inflectional information: s (singular), p (plural), m (masculine), f (feminine), n (neutral), W (infinitive), P (participle), G (gerund), P1s (1^a person singular of present indicative),...

Following this procedure, we elaborated a total of 192 FST graphs that include the inflectional variants of simple words, expressions, specialized terms, place names, terms in Latin and abbreviations or acronyms. To arrive at the inflection of the words, we first distinguish the different classes or general categories such as N (noun), V (verb) and A (adjective).

Inflectional analysis thus requires the previous consideration of two fundamental structures: the stem or base form of a word, and the inflectional affixes. The inflectional

forms of words will belong to a closed system or inflectional paradigm (Matthews, 1965, 1974) that contains an ordered enumeration of each form that a stem may present. The term paradigm is understood here as the set of all forms that serve as a model of inflection of a certain word class. When paradigmatic relationships prevail, the given forms can be inserted in the same word position (Chomsky, 1957).

The structure of the paradigm refers to the number of grammatical categories that may appear in the interior of the paradigms (Coseriu, 1981). The set of elements constituting the paradigm has a constant value represented by the stem and different intracategorical codes indicated by inflection. The intracategorical oppositions in the interior of the inflectional paradigm are organized on the basis of the different grammatical categories. In turn, depending on that number, we may speak of paradigms with simple structures (when only one dimension or category is involved) and complex structures (when several categories are involved).

On the other hand, the term variants involve transformations in the internal structure of a word, of either an inflectional or derivational nature. The changes produced by inflectional variation never alter the POS category of a word, and hardly affect meaning. Yet the modifications of a derivational nature often imply different POS tagging as well as a change in the word class and meaning. Because the purpose of conflation methods is to match non-identical words that refer to the same main concept, we do not deal with derivational affixes in this application. A later section addresses the formal aspects of the representation of the structures of these different inflectional paradigms.

Nominal inflection paradigms

The nominal inflection paradigm has oppositions organized in terms of the categories gender/number. Within gender we have masculine/feminine, and neutral; and for number we have singular/plural, as shown in Figure 4.

The nouns belonging to one same inflectional paradigm are marked with the same POS tag and the same numerical code, to which a graphic transducer is linked; this groups all the word classes that are inflected in the same way. The irregularities are represented directly in FST graphs, bypassing the morphological rule component.

With the technique described above we developed a total of 33 paradigms of nominal inflection represented in the graphic transducers: FST (N, N1, N10, N101, N102, N103, N11, N12, N13, N14, N15, N16, N2, ...).

Adjectival inflection paradigms

In Spanish, the adjective normally works as an adjunct nominal complement, and the number of ways it can appear depends on the forms that the noun may present.

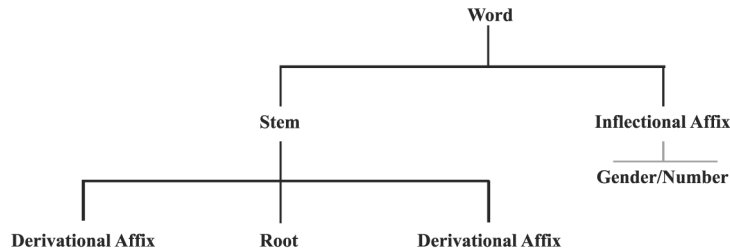


Figure 4.
Nominal inflection
paradigm

The most simple formalization of the adjectives presents a minimum of two forms in the opposition of gender, masculine/feminine, and two more regarding number, singular/plural, bringing us to a maximum of four forms in the opposition of gender/number.

In order to formalize all the inflected variants, we elaborated 27 paradigms of adjectival inflection represented in a graphic FST, where the different classes of sets are possessives ordinals, non-numerical quantifiers, or participles as adjectives. Thus, we arrive at the graphic transducers of adjectival inflection: FST (A, A1, A2, A3, A4, A5, A6, A7, A8, A9, . . .).

Verbal inflection paradigms

The most highly inflected element in the romance languages is the verb, and for this reason there are many studies of its inflectional paradigm (Matthews, 1974; Alcoba, 1991; Harris, 1987; Ambadiang, 1990, 1994; Mighetto, 1992). Verbs feature a binary structure of stem and inflectional forms. One fundamental element in this organization is the vowel of the stem, which determines the classification of the verb into a particular conjugation group. The vowel of the stem (-a, -e, -i) signals the conjugation (1^a 2^a 3^a), and is transformed accordingly in the endings, on the basis of tense/mode/aspect and number/person. Harris (1987) represents the structure of the verbal constituents according to the model (Figure 5).

The category tense would be specified as present/preterite/future/conditional. The category mode, reflecting the certainty of the verbal expressions, would be indicative/subjunctive/imperative. Meanwhile, aspect refers to the stage of development of the action, in the binary category of perfect/imperfect. Additionally, for the constituents number/person, we have singular/plural and the inflectional forms for 1^a 2^a 3^a person. All the verbal inflection forms, therefore, are assigned to the multiple combinations allowed by the inflectional constituents of tense/mode/aspect and number/person, except for the non-personal forms (infinitive, gerund and past participle).

The problem is that the flectional forms of the respective conjugations produce variants and irregularities in the combinations. The identification and posterior representation of the alternations in verbal inflection is quite laborious, as many times it requires the conjugation to be considered individually for each irregular inflection. The procedure we followed was to place all the verbs into one of the three major groups of verbal conjugation according to the vowel of the stem. Then, in each group we

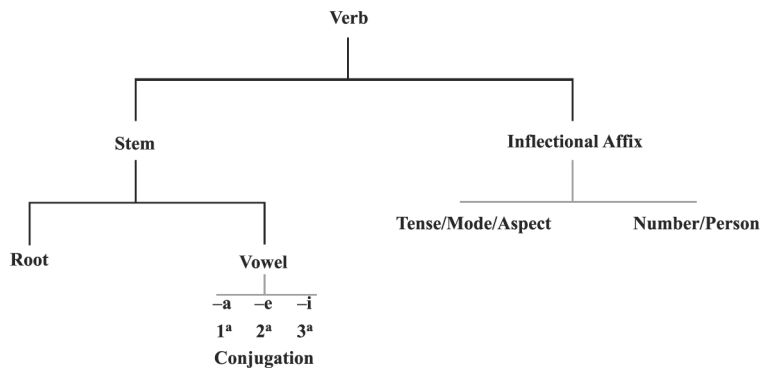


Figure 5.
Verbal inflection
paradigm

distinguish those belonging to regular paradigms and those that have irregular paradigms. It is understood that the regular verbs have invariable stems in all the inflected forms of conjugation. The irregular verbs have roots that are partly or fully altered in the inflected forms. It is complicated to represent the irregular inflections because they often require case by case processing.

We grouped all the regular conjugations into the three major paradigms of regular inflection first, and then went on to identify the different classes of irregular inflectional paradigms. This system led us to a total of 99 paradigms of verbal inflection, classified as:

- (1) Paradigm of regular inflection of the 1st conjugation: FST V1:
 - Paradigms of irregular inflection of the 1st conjugation: FST (V10, V101, V102, V103, V104, V105, V106, V107, V108, V109, V11, V110,...).
- (2) Paradigm of regular inflection of the second conjugation: FST V2:
 - Paradigms of irregular inflection of the second conjugation: FST (V20, V201, V202, V203, V204, V205, V206, V207, V208, V209, V21, V210,...).
- (3) Paradigm of regular inflection of the third conjugation: FST V3:
 - Paradigms of irregular inflection of the 3rd conjugation: FST (V30, V301, V302, V303, V304, V305, V306, V307, V308, V309, V31, V310,...).

Evaluation

For the purpose of evaluating conflation performance, we collected a test corpus, extracted from bibliographic records randomly selected from a search in the ISOC database of the *Consejo Superior de Investigaciones Científicas* (CSIC), which includes references of periodical publications in Spanish within the area of information science. The collection contained 18,215 words, to which two processes of standardization were applied: electronic dictionaries and a Spanish version of the Porter algorithm generated with the Snowball project[2]. Previous to evaluation, different pre-processing stages of the test corpus were needed, depending on the conflation method.

Lemmatization – pre-processing phase

Before applying lemmatization tools, the test corpus is transformed into a text file with the ASCII format, in which strings of characters are divided up into sequences of units between blank spaces or tokens. This phase leaves us with:

- Identification and segmentation of the elements that reflect the logical structure of the text, such as paragraphs, sentences or punctuation marks, by inserting delimitation marks at the end of each logical element. For the segmentation of these logical units of analysis we designed a graphic transducer in charge of inserting a marker of surface delimitation, {S}, at the end of every paragraph, after all periods and semicolons, or after any delimiter.
- Recognition of the non-autonomous compound forms, such as frozen or idiomatic expressions made up of several words with a fixed form, using the DELACF dictionary.
- Identification of contracted words, such as terms that cannot be assigned to any single category because they are formally equivalent to two successive categories. For instance, “del,” which is a contraction of the article “el” and the

preposition “de.” To analyze such words we built graphic transducers that, given a contracted form, produce an output corresponding to the decomposition of the contraction into its base constituents.

In addition, we performed basic statistics on the texts, allowing us to count the number of occurrences of each token, and then sort them according to their frequency. In Table II, we show the breakdown of these units: the lexical units linked to linguistic information, the digits, the delimiters (such as periods, commas, semicolons, or hyphens) and lines or sentences. The final test collection contained 3,296 words.

Once the sentences of the collection of verification are pre-processed, the next step is to identify the lemmas of the lexical units using the electronic dictionaries developed with finite-state technology: the DELAF-type dictionary and DELACF-type dictionary. The composition of the electronic dictionaries is specified in Table III (the total number of dictionary entries is not very high due to the fact that we did not build wide-coverage lemmatizing tools).

By applying the lexical resources, we obtain two transformations of the lexical units of the corpus that provide linguistic information of two sorts. First, the reduction of all utterances of inflected forms to canonical forms; and second, the assignment of POS tags to all the lexical units. At the same time, the result of the two processes can be presented in different modes in the lineal tagging:

- *Tagging of the lexical units in: {lemma + POS tag}*. Here all the lemmatized lexical forms correctly grouped in a single lemma and a single POS category are recognized.
- *Tagging the lexical units in: {inflected form + POS tag}*. Here the total inflected lexical forms are recognized, as well as the possible lexical variants, with POS categories.
- *Analysis of the lexical units in: {lemma}*. In this analysis, the total of the lemmatized lexical forms grouped on canonical forms are recognized, although the lemma may have different POS categories assigned to it.

	Tokens	Different tokens
Lexical units	18,215	3,296
Digits	4,082	10
Delimiters	5,508	16
Lines	1,676	
Total number of tokens	27,805	3,322

Table II.
Composition of the test
corpus

Expanded dictionary of inflected forms (DELAF)	60,511
Dictionary of compounds (DELACF)	1,148
Total number of dictionary entries	61,659

Table III.
Composition of the
electronic dictionaries

Stemming – pre-processing phase

Before applying the stemmer it is necessary to carry out a stage of preprocessing because this algorithm was originally devised for the conflation of words in English. In the case of Spanish in particular, this algorithm presents a series of deficiencies in its results, as the presence/absence of accents and tildes can imply separate meanings. Indeed, in order to apply Porter's stemmer we had to remove these symbols so that it would perform as expected. Moreover, at this stage we eliminated duplicate words, common words and a stopword list, transferred from the Snowball list[2], expanding them into all possible variants. After this, the test collection contained 1,930 unique words.

Metrics of evaluation

To calculate conflation correctness, we used an adaptation of recall and precision measures. Recall would indicate the proportion of terms that are conflated with respect to a set of sequences of evaluation. We shall redefine it as correct variants conflated over total possible variants susceptible of lemmatization, or stemming. The precision could be redefined as the ratio of valid variants conflated from among the total variants identified by the lemmatizer, or stemmer. The two parameters were calculated using the following equations:

$$\text{Recall}(R) = \frac{\text{Number of correct variants lemmatized/Stemmed}}{\text{Total number of possible variants}}$$

$$\text{Precision}(P) = \frac{\text{Number of correct variants lemmatized/Stemmed}}{\text{Total number of words lemmatized/Stemmed}}$$

In addition, a measure of performance was used that takes into account both recall and precision, the F -measure (van Rijsbergen, 1979) – defined as the harmonic mean of recall and precision, as compared to the arithmetic mean – which exhibits the desirable properties of being highest when both recall and precision are high (where β is a pre-established value, and if $\beta = 1$ it means that recall and precision are equally weighted ($R = P$); whereas $\beta > 1$ means more weight to recall, and $\beta < 1$ means precision weighs more). They were calculated as:

$$F_{\beta} = \frac{(\beta^2 + 1)RP}{\beta^2R + P}$$

To assess these parameters for the lemmatizer we need to obtain the following data of frequency:

- *Number of correct variants lemmatized.* For these occurrences, we opted to apply the lexical analyzers in the mode {lemma + POS tag}, where each word of the *corpus* is automatically grouped or related with its corresponding lemma and POS category. We eliminate underanalysis errors (words unlemmatized) and overanalysis errors (nonvariants that are lemmatized).
- *Total number of possible variants.* To arrive at this data, we applied the lexical analyzers in the mode {inflected form + POS tag}, so that each word of the *corpus* is identified with its inflected variants and their POS category.

- *Total number of variants lemmatized.* To obtain this occurrence figure, we applied the lexical analyzers in the mode {lemma}, in which each word of the *corpus* is grouped or related with its corresponding lemma. These data would be obtained by subtracting the number of unlemmatized variants from the total unique words.

In the case of the stemmer, in order to calculate these measures, it was necessary to acquire the following data of frequency by manually comparing output words:

- *Number of correct variants stemmed.* For these occurrences, we compared the stemmer's output to its input and identified the words that had been successfully merged by the conflation procedure, removing understemming and overstemming errors.
- *Total number of possible variants.* To arrive at this data, we identify the total number of variants that should have been grouped to a given stem. This was done manually. Then we eliminated terms like proper names, compound words, initials, abbreviations, spelling errors, terms in foreign languages, and general terms that are not authentic variants.
- *Total number of variants stemmed.* To obtain this occurrence information, we proceeded to subtract from the total number of unique words, the number of variants not stemmed (understemming errors), leaving the total number of variants stemmed.

The proportion of underanalysis/understemming and overanalysis/overstemming errors could be calculated as follows:

$$\begin{aligned} & \text{Underanalysis/Understemming errors} \\ & = \frac{\text{Number of variants not lemmatized/Stemmed}}{\text{Total number of possible variants}} \end{aligned}$$

$$\begin{aligned} & \text{Overanalysis/Overstemming errors} \\ & = \frac{\text{Number of non-variants lemmatized/Stemmed}}{\text{Total number of words lemmatized/Stemmed}} \end{aligned}$$

Results and discussion

The results of evaluation and error percentages are presented in Tables IV and V, respectively. A look at the recall of the lexical analyzers shows that they recognize and correctly lemmatize 73.6 percent of the possible lexical variants. Over a total of 2,216 possible variants, then, we would obtain a correct grouping of 1,632. The number of unlemmatized forms is 1,607, not including unknown terms (mainly consisting of spelling errors, proper names or foreign words). Results in the case of recall are moderately good, as with respect to the *F*-measure, the scores are under 83 percent. In consequence, the greatest weakness of lexical analyzers developed with finite-state technology would be their inability to standardize certain units. When evaluating the results, we calculated over lemmatized lexical variants. Yet when there is a problem of ambiguity, the variants are not lemmatized. The reason is that, in linear tagging, the

Table IV.
Evaluation results

	Lemmatization	Stemming
Total number of unique words	3,296	1,930
Total number of words lemmatized/stemmed	1,689	1,702
Number of correct variants lemmatized/stemmed	1,632	1,314
Total number of possible variants	2,216	1,357
Number variants not lemmatized/stemmed	1,607	228
Number of nonvariants lemmatized/stemmed	57	388
Recall	<i>0.73</i>	<i>0.97</i>
Precision	<i>0.96</i>	<i>0.77</i>
F_1	<i>0.83</i>	<i>0.86</i>

Table V.
Error percentages

	Lemmatization	Stemming
Underanalysis/understemming errors	0.72	0.16
Overanalysis/overstemming errors	0.04	0.23

lexical units that can be associated with more than one canonical form are not reduced or given POS tags. For instance, the variant “modelo” could feasibly be mapped to different lemmas {modelo,modelo.N4:ms} or {modelo,modelar.V1:P1s}. The crucial problem of NLP-oriented conflation methods is, therefore, underanalysis. This deficiency is well-known in the literature, as such tools cannot reduce variants when they can be conflated to more than one standardized form, unless disambiguation methods are used.

Precision of the analyzers is very high, 96.6 percent, with F -measure scores exceeding 83 percent. Any failure owes to cases where two or more variants apparently correspond to a single lemma, yet are actually distinct variants because they have different POS tags. For instance, “scientific” and “scientist” have the same surface and canonical forms in Spanish, so that {cientifico,cientifico.A1:ms} {cientifico,cientifico.N1:ms}. The overanalysis error percentage was found to be 0.04. In turn, a minor flaw of the system is that lexical variants may be incorrectly lemmatized, causing certain variants to be mistakenly linked to the same canonical form, when in fact they are distinct lemmas, each with a different POS tag. Such failures in precision could be improved by always applying the lexical analyzers in the mode {lemma + POS tag}, which would allow us to eliminate the inaccuracies caused by the grouping of two or more variants to a single lemma when they are actually different variants.

From the stemming evaluation, the recall result on the test corpus was remarkably high, 97 percent, with F -measure scores exceeding 86 percent. Over a total of 1,357 possible variants, the stemmer correctly grouped 1,314. We evaluated the stemmer by manually computing the number of words represented by the expected stem. For instance, consider the Spanish words “digital,” “digitalizar,” “digitalizacion.” Applying the stemmer, we obtain “digital” and “digitaliz,” which leaves us with understemming errors, since the three words are not conflated with the same stem.

On the other hand, the results for precision are not very high, 77 percent, with F -measure scores under 86 percent. In this case, over a total of 1,702 words merged, the stemmer correctly grouped 1,314. To illustrate, take the Spanish word “parte,” which

comes from the verb “partir” (“to depart” in English) and the noun “parte” (“part” in English). By means of the stemmer, we obtain for both “part,” meaning an overstemming error, since the two words should have been grouped to different stems. Our finding was a comparatively high percentage of overstemmed words, 0.23. This is a fairly common matter in the sphere of this type of process, because the POS category to which the words belong is not taken into account, and so the overstemming errors are the main cause of failures. In addition, a great number of nonvariants were conflated by the stemmer: foreign words, proper names, abbreviations and acronyms.

Conclusions

The application of lexical databases to a test corpus provides assessments of the accuracy of lexical analyzers built using finite-state technology for processing lexical variants in Spanish. First, the tools of standardization developed present a moderate to high percentage of recall. Second, precision is very high due to the small percentage of overanalysis errors. Third, the main inconsistency of these analyzers is underanalysis; that is, they standardize only those variants that can be mapped to a single controlled form, and therefore in a situation of ambiguity they do not lemmatize. If, however, we built task-oriented and specialized dictionaries to work for small vocabularies, the possibility of a single variant having different valid analyses would be largely reduced.

We can state in summary that these comparative experiments reveal as the main weakness of NLP-oriented methods the high percentage of words underanalyzed, and a noteworthy strength in the low percentage of words overanalyzed. In contrast, the main weakness of stemming methods is the high percentage of words overstemmed, and their main strength is the low percentage of words understemmed. Bearing in mind that underanalysis/understemming errors affect recall, whereas overanalysis/overstemming errors affect precision, we can conclude that finite-state methods are more accurate than stemming procedures, and would consequently prove more precise for normalization processes. This feature would make them especially adequate for other subsequent applications such as recognition and indexing of noun phrases, conflation of multiterms, pattern-matching, or IE.

Notes

1. www.searchtools.com/tools/muscat.html (site visited July 2005).
2. <http://snowball.tartarus.org/algorithms/spanish/stemmer.html> (site visited July 2005).

References

- Adamson, G.W. and Boreham, J. (1974), “The use of an association measure based on character structure to identify semantically related pairs of words and document titles”, *Information Storage and Retrieval*, Vol. 10 No. 1, pp. 253-60.
- Alcoba, S. (1991), “Morfología del verbo español”, in Martin Vide, C. (Ed.), *Lenguajes Naturales y Lenguajes Formales*, Publicaciones de la Universidad, Barcelona.
- Allan, J., Ballesteros, L., Callan, J.P., Croft, W.B. and Lu, Z. (1995), “Recent experiments with INQUERY”, in Harman, D.K. (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, Institute of Standards and Technology Special Publication 500-236, Gaithersburg, MD, pp. 49-63.
- Allan, J., Callan, J.P., Croft, B.W., Ballesteros, L., Broglio, J., Xu, J. and Shu, H. (1996), “INQUERY at TREC-5”, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*,

- National Institute of Standards and Technology Special Publication 500-238, Gaithersburg, MD, pp. 119-32.
- Ambadiang, T. (1990), "Contribución al estudio del verbo español: un análisis morfosemántico", *Anuario de Lingüística Hispánica*, Vol. 6, pp. 29-63.
- Ambadiang, T. (1994), *La Morfología Flexiva*, Taurus, Madrid.
- Angell, R.C., Freund, G.E. and Willett, P. (1983), "Automatic spelling correction using a trigram similarity measure", *Information Processing & Management*, Vol. 19 No. 4, pp. 255-61.
- Antworth, E.L. (1995), "User's guide to PC-KIMMO Version 2", available at: www.sil.org/pckimmo/v2/doc/guide.html
- Broglio, J., Callan, J.P., Croft, W.B. and Nachbar, D.W. (1994), "Document retrieval and routing using the INQUERY system", in Harman, D.K. (Ed.), *Proceedings of the Third Text REtrieval Conference (TREC-3)*, National Institute of Standards and Technology Special Publication 500-225, Gaithersburg, MD, pp. 29-38.
- Buckley, C., Salton, G., Allan, J. and Singhal, A. (1994), "Automatic query expansion using SMART: TREC 3", in Harman, D.K. (Ed.), *Proceedings of the Third Text REtrieval Conference (TREC-3)*, National Institute of Standards and Technology Special Publication 500-225, Gaithersburg, MD, pp. 69-80.
- Buckley, C., Singhal, A. and Mitra, M. (1996), "Using query zoning and correlation within SMART: TREC 5", in Harman, D.K. (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-5)*, National Institute of Standards and Technology Special Publication 500-238, Gaithersburg, MD, pp. 105-18.
- Buckley, C., Singhal, A., Mitra, M. and Salton, G. (1995), "New retrieval approaches using SMART: TREC 4", in Harman, D.K. (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, MD, pp. 25-48.
- Carmona, J., Cervell, S., Márquez, L., Martí, M.A., Padró, L., Placer, R., Rodríguez, H., Taulé, M. and Turmo, J. (1998), "An environment for morphosyntactic processing of Spanish unrestricted text", paper presented at First International Conference on Language Resources and Evaluation, LREC'98, Granada, pp. 915-22.
- Cavnar, W.B. (1994), "Using an *n*-gram based document representation with a vector processing retrieval model", in Harman, D.K. (Ed.), *Proceedings of the Third Text REtrieval Conference (TREC-3)*, National Institute of Standards and Technology, Gaithersburg, MD, pp. 269-78.
- Chomsky, N. (1957), *Syntactic Structures*, Mouton, The Hague.
- Chomsky, N. and Halle, M. (1968), *The Sound Pattern of English*, Harper and Row, New York, NY.
- Coseriu, E. (1981), *Lecciones de Lingüística General*, Gredos, Madrid.
- Damashek, M. (1995), "Gauging similarity with *n*-grams: language independent categorization of text", *Science*, Vol. 267, pp. 843-8.
- Dawson, J.L. (1974), "Suffix removal for word conflation", *Bulletin of the Association for Literary and Linguistic Computing*, Vol. 2 No. 3, pp. 33-46.
- Figuerola, C.G., Gómez, R., Zazo Rodríguez, A.F. and Alonso Berrocal, J.L. (2002), "Stemming in Spanish: a first approach to its impact on information retrieval", in Peters, C., Braschler, M., Gonzalo, J. and Kluck, M. (Eds) paper presented at Evaluation of Cross-language Information Retrieval Systems, Second Workshop of the Cross-language Evaluation Forum, CLEF 2001, Springer-Verlag, Berlin, (*Lecture Notes in Computer Science*, Vol. 2406).

-
- Frakes, W.B. (1992), "Stemming algorithms", in Frakes, W.B. and Baeza-Yates, R. (Eds), *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.
- Frakes, W.B. and Fox, C.J. (2003), "Strength and similarity of affix removal stemming algorithms", *ACM SIGIR Forum*, Vol. 37 No. 1, pp. 26-30.
- Galvez, C., Moya-Anegón, F. and Solana, V.H. (2005), "Term conflation methods in information retrieval: non-linguistic and linguistic approaches", *Journal of Documentation*, Vol. 61 No. 4, pp. 520-47.
- Gey, F.C., Chen, J.A., He, M. and Jason, M. (1995), "Logistic regression at TREC 4: probabilistic retrieval from full text document collections", in Harman, D.K. (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, MD, pp. 65-72.
- Graña, J., Barcala, F.M. and Alonso, A. (2001), "Compilation methods of minimal acyclic automata for large dictionaries", in Watson, B.W. and Wood, D. (Eds), *Proceedings of the Sixth Conference on Implementations and Applications of Automata (CIAA 2001)*, Pretoria, South Africa, pp. 116-29.
- Harman, D.K. (1991), "How effective is suffixing?", *Journal of the American Society for Information Science*, Vol. 47 No. 1, pp. 70-84.
- Harris, J.W. (1987), "The accentual patterns of verb paradigms in Spanish", *Natural Language and Linguistic Theory*, Vol. 5, pp. 61-90.
- Hearst, M., Pedersen, J.O., Pirolli, P., Schütze, H., Grefenstette, G. and Hull, D.A. (1995), "Xerox site report: four TREC-4 tracks", in Harman, D.K. (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, MD, pp. 97-119.
- Hull, D.A. (1996), "Stemming algorithms: a case study for detailed evaluation", *Journal of the American Society for Information Science*, Vol. 47 No. 1, pp. 70-84.
- Hull, D.A., Grefenstette, G., Schulze, B.M., Gaussier, E., Schütze, H. and Pedersen, J.O. (1996), "Xerox TREC-5 site report: routing filtering, NLP and Spanish tracks", in Voorhees, E.M. and Harman, D.K. (Eds), *The Fifth Text Retrieval Conference (TREC-5)*, National Institute of Standards and Technology Special Publication 500-238, Gaithersburg, MD, pp. 167-80.
- Jacquemin, C. and Tzoukermann, E. (1999), "NLP for term variant extraction: synergy between morphology, lexicon, and syntax", in Strzalkowski, T. (Ed.), *Natural Language Information Retrieval*, Kluwer, Dordrecht.
- Johnson, C.D. (1972), *Formal Aspects of Phonological Description*, Mouton, The Hague.
- Kaplan, R.M. and Kay, M. (1981), "Phonological rules and finite-state transducers", paper presented at Linguistic Society of America Meeting Handbook, Fifty-sixth Annual Meeting, New York, NY.
- Karttunen, L. (1983), "KIMMO: a general morphological processor", *Texas Linguistics Forum*, Vol. 22, pp. 217-28.
- Karttunen, L. (1994), "Constructing lexical transducers", *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, pp. 406-11.
- Karttunen, L., Kaplan, R.M. and Zaenen, A. (1992), "Two-level morphology with composition", *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*, Nantes, France, pp. 141-8.
- Kelley, F. and Smeaton, A.F. (1996), "TREC-5 experiments at Dublin City University: query space reduction, Spanish stemming and character shape encoding", in Voorhees, E.M. and Harman, D.K. (Eds), *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*,

National Institute of Standards and Technology Special Publication 500-238, Gaithersburg, MD, pp. 57-64.

- Kosinov, S. (2001), "Evaluation of n -grams conflation approach in text-based information retrieval", *Proceedings of International Workshop on Information Retrieval, Oulu, Finland*.
- Koskenniemi, K. (1983), *Two-level Morphology: A General Computational Model for Word-form Recognition and Production*, Department of General Linguistics, University of Helsinki, Helsinki.
- Kraaij, W. and Pohlmann, R. (1994), "Porter's stemming algorithm for Dutch", in Noordman, L.G.M. and de Vroomen, W.A.M. (Eds) paper presented at Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie, Tilburg, pp. 167-80.
- Kraaij, W. and Pohlmann, R. (1995), "Evaluation of a Dutch stemming algorithm", in Rowley, R. (Ed.), *The New Review of Document and Text Management*, Vol. 1, Taylor Graham, London.
- Krovetz, R. (1993), "Viewing morphology as an inference process", in Korfhage, R., Rasmussen, E.M. and Willett, P. (Eds), *Proceedings of the 16th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, pp. 191-202.
- Lennon, M., Pierce, D.S., Tarry, B.D. and Willett, P. (1981), "An evaluation of some conflation algorithms for information retrieval", *Journal of Information Science*, Vol. 3 No. 4, pp. 177-83.
- Lovins, J.B. (1968), "Development of a stemming algorithm", *Mechanical Translation and Computational Linguistics*, Vol. 11, pp. 22-31.
- Matthews, P.H. (1965), "The inflection component of a word-and-paradigm grammar", *Journal of Linguistics*, Vol. 1, pp. 139-71.
- Matthews, P.H. (1974), *Morphology: An Introduction to the Theory of Word-structure*, Cambridge University Press, Cambridge, MA.
- Mighetto, D. (1992), "Notas sobre la noción de aspecto en un marco de clasificación de verbos (Vb) y sustantivos verbales (Sv)", *Voz y Letra*, Vol. 3 No. 1, pp. 69-100.
- Nakov, P. (2003), "Building an inflectional stemmer for Bulgarian", *Proceedings of Fourth International Conference on Computer Systems and Technologies (ICCST'03)*, ACM Press, New York, NY, pp. 419-24.
- Paice, C.D. (1990), "Another stemmer", *ACM SIGIR Forum*, Vol. 24 No. 3, pp. 56-61.
- Paice, C.D. (1996), "A method for evaluation of stemming algorithms based on error counting", *Journal of the American Society for Information Science*, Vol. 47 No. 8, pp. 632-49.
- Pirkola, A. (2001), "Morphological typology of languages for IR", *Journal of Documentation*, Vol. 57 No. 3, pp. 330-48.
- Popovic, M. and Willett, P. (1992), "The effectiveness of stemming for natural-language access to Slovene textual data", *Journal of the American Society for Information Science*, Vol. 43 No. 5, pp. 384-90.
- Porter, M.F. (1980), "An algorithm for suffix stripping", *Program*, Vol. 14, pp. 130-7.
- Robertson, A.M. and Willett, P. (1998), "Applications of n -grams in textual information systems", *Journal of Documentation*, Vol. 54 No. 1, pp. 48-69.
- Rodríguez, S. and Carretero, J. (1996), "A formal approach to Spanish morphology: the COES tools", paper presented at XII Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN), SEPLN, Sevilla, pp. 118-26.

-
- Savoy, J. (1993), "Stemming of French words based on grammatical categories", *Journal of the American Society for Information Science*, Vol. 44 No. 1, pp. 1-9.
- Silberztein, M. (1993), *Dictionnaires Électroniques et Analyse Automatique de Textes: le Système INTEX*, Masson, Paris.
- Silberztein, M. (2000), "INTEX: an FST toolbox", *Theoretical Computer Science*, Vol. 231 No. 1, pp. 33-46.
- Sparck Jones, K. and Tait, J.I. (1984), "Automatic search term variant generation", *Journal of Documentation*, Vol. 40 No. 1, pp. 50-66.
- Tzoukermann, E., Klavans, J.L. and Jacquemin, C. (1997), "Effective use of natural language processing techniques for automatic conflation of multi-word terms: the role of derivational morphology, part of speech tagging, and shallow parsing", *Proceedings of 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, Philadelphia, PA, Vol. 97, pp. 148-55.
- van Rijsbergen, C.J. (1979), *Information Retrieval*, Butterworths, London.
- Vilares, J., Alonso, M.A., Ribadas, F.J. and Vilares, M. (2003), "COLE experiments at CLEF 2002 Spanish monolingual track", in Peters, C., Braschler, M., Gonzalo, J. and Kluck, M. (Eds), *Advances in Cross-language Information Retrieval*, Springer-Verlag, Berlin, (*Lecture Notes in Computer Science*, Vol. 2785), pp. 265-78.
- Voutilainen, A. (1995), "Morphological disambiguation", in Karlsson, F., Voutilainen, A., Heikkilä, J. and Anttila, A. (Eds), *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*, Mouton de Gruyter, Berlin, pp. 165-284.
- Xu, J. and Croft, B. (1998), "Corpus-based stemming using co-occurrence of word variants", *ACM Transactions on Information Systems*, Vol. 16 No. 1, pp. 61-81.

Corresponding author

Carmen Galvez can be contacted at: cgalvez@ugr.es