UFRGS@CLEF2008: Using Association Rules for Cross-Language Information Retrieval

André Pinto Geraldo, Viviane Moreira Orengo
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
[apgeraldo,vmorengo]@inf.ufrgs.br

Abstract

For UFRGS's participation on the TEL task at CLEF2008, our aim was to assess the validity of using algorithms for mining association rules to find mappings between concepts on a Cross-Language Information Retrieval scenario. Our approach requires a sample of parallel documents to serve as the basis for the generation of the association rules. The results of the experiments show that the performance of our approach is not statistically different from the monolingual baseline in terms of mean average precision. This is an indication that association rules can be effectively used to map concepts between languages. We have also tested a modification to BM25 that aims at increasing the weight of rare terms. The results show that this modified version achieved better performance. The improvements were considered to be statistically significant in terms of MAP on our monolingual runs.

Categories and Subject Descriptors

H.3.1 [Content Analysis and Indexing]: Linguistic processing. H.3.4 [Systems and Software]: Performance evaluation

Free Keywords

association rules, experimentation, performance measurement

1 Introduction

This paper reports on monolingual and bilingual ad-hoc information retrieval experiments that we have performed for the TEL task at CLEF2008. Our aim was to use algorithms for mining association rules to map concepts between languages, on a Cross-Language Information Retrieval (CLIR) scenario. These algorithms are widely used for data mining purposes. A common example is market-basket data, i.e. the items that a customer buys at one transaction. For such data, an association rule would state, for example, that "90% of customers that purchase bread also purchase milk".

The motivation is that such algorithms are computationally cheaper than other co-occurrence-based techniques such as Latent Semantic Indexing (Deerwester et al., 1990). Our goal was to use automatic methods that did not employ resources such as dictionaries, thesauri or machine translation.

The remainder of this paper is organised as follows: Section 2 proposes an approach for using algorithms for mining association rules for CLIR; Section 3 presents some modifications we implemented on the Okapi BM25 formula to improve retrieval results; Section 5 discusses the experiments and results; and Section 6 presents the conclusions.

2 Association Rules for CLIR

An association rule (AR) is an implication of the form $X \Rightarrow Y$, where $X = \{x_1,x_2,\ldots,x_n\}$, and $Y = \{y_1,y_2,\ldots,y_m\}$ are sets of items. The problem of mining ARs in market-basket data was firstly investigated by (Agrawal et al., 1993). In the rule "90% of customers that purchase bread also purchase milk", the antecedent is bread and the consequent is milk. The number 90% is the confidence factor (*conf*) of the rule, which is calculated according to equation 1. The confidence of the rule can be interpreted as the

probability that the items in the consequent will be purchased given that the items in the antecedent are purchased.

$$conf(X \Rightarrow Y) = \frac{n(X \bigcup Y)}{n(X)} \quad (1)$$

where n is the number of transactions.

An AR also has a support level associated to it. The support (*sup*) of a rule refers to how frequently the sets of items $X \cup Y$ occur in the database. In other words, *sup* expresses the percentage of the transactions that contain all items in X and Y. Equation 2 shows how the support of an AR is calculated.

$$\sup(X \Rightarrow Y) = \frac{n(X \bigcup Y)}{N} \quad (2)$$

where N is the total number of transactions in the database.

The problem of mining ARs is to generate all rules that have support and confidence greater than predefined thresholds. We have used the Apriori Algorithm (Agrawal & Srikant, 1994) to extract the ARs. Figure 1 shows the execution of Apriori for the items *a*, *b*, *c*, *d* and *e*. The algorithm calculates the support of the individual items and then proceeds by combining the individual items two-by-two, three-by-three and so on. If the support of the itemset is lower than the threshold *minsup*, this itemset is discarded. More formally, let I be an itemset, for each subset $v \subseteq I$ the algorithm will generate a rule of the form $v \Rightarrow (I - v)$ if *sup(I)/sup(v)* is greater than *minsup*.

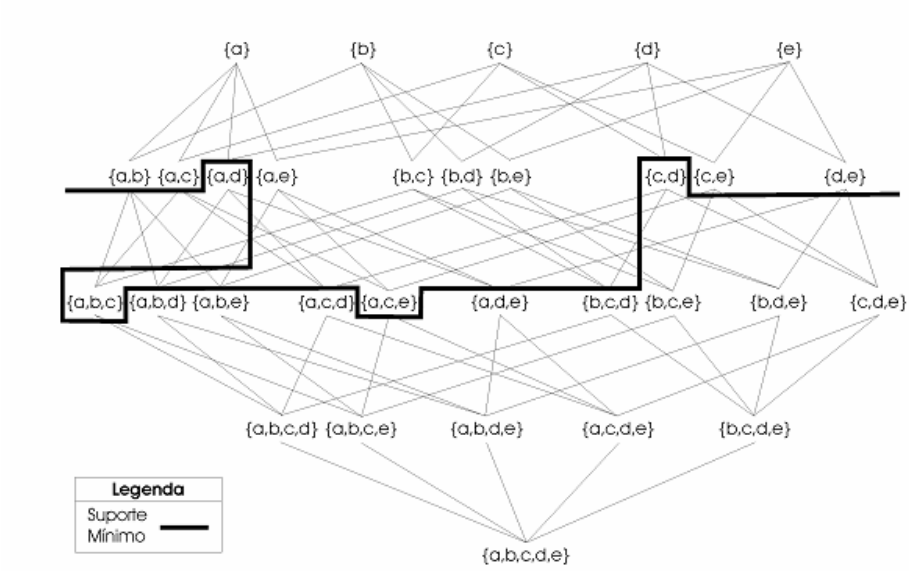

Figure 1 – Apriori execution (Hipp & Güntzer, 2002)

Our proposal is to map the problem of finding ARs between items in a market-basket scenario to the problem of finding cross-linguistic equivalents between a pair of languages on a parallel corpus. This approach is based on co-occurrences and works under the assumption that cross-linguistic equivalents would have a significant number of co-occurrences over a parallel corpus. In our approach, the transaction database is replaced by a text collection; the items that the customer buys correspond to the terms in the text; and the shopping transactions are represented by documents.

The proposed approach to use algorithms for mining ARs for CLIR can be divided into five phases: (i) pre-processing, (ii) mining ARs, (iii) rule filtering, (iv) query translation, and (v) query execution. Figure 2 depicts this process. Next we explain each phase.

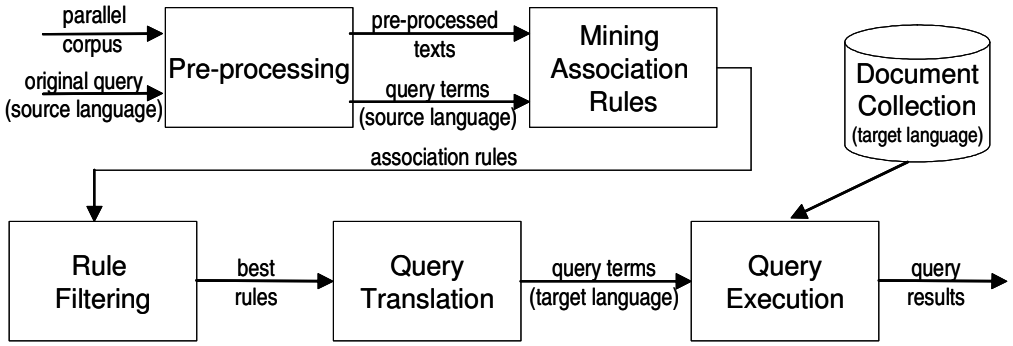

Figure 2 – Proposed approach for using association rules for CLIR

i. **Pre-processing**: The inputs for this phase are a collection of parallel documents and the original query in the source language. During this phase the original text in a language and its equivalent in the other language are initially treated separately. We remove stop-words, apply stemming, break the documents into sentences, and tag all terms in one of the languages with a prefix (e.g. all English words are tagged with an "E="). The aim of the tagging is to avoid generating rules between words in the same language. The last step is to merge each sentence with its translation. The output of this phase is a set of pre-processed parallel sentences. During this phase, an inverted index containing all stems in the document collection and the list of sentences in which they appear is also built. The inverted index will be used in the next phase to enable selection of the sentences over which the Apriori algorithm is run. The pre-processing phase is shown in Figure 3.

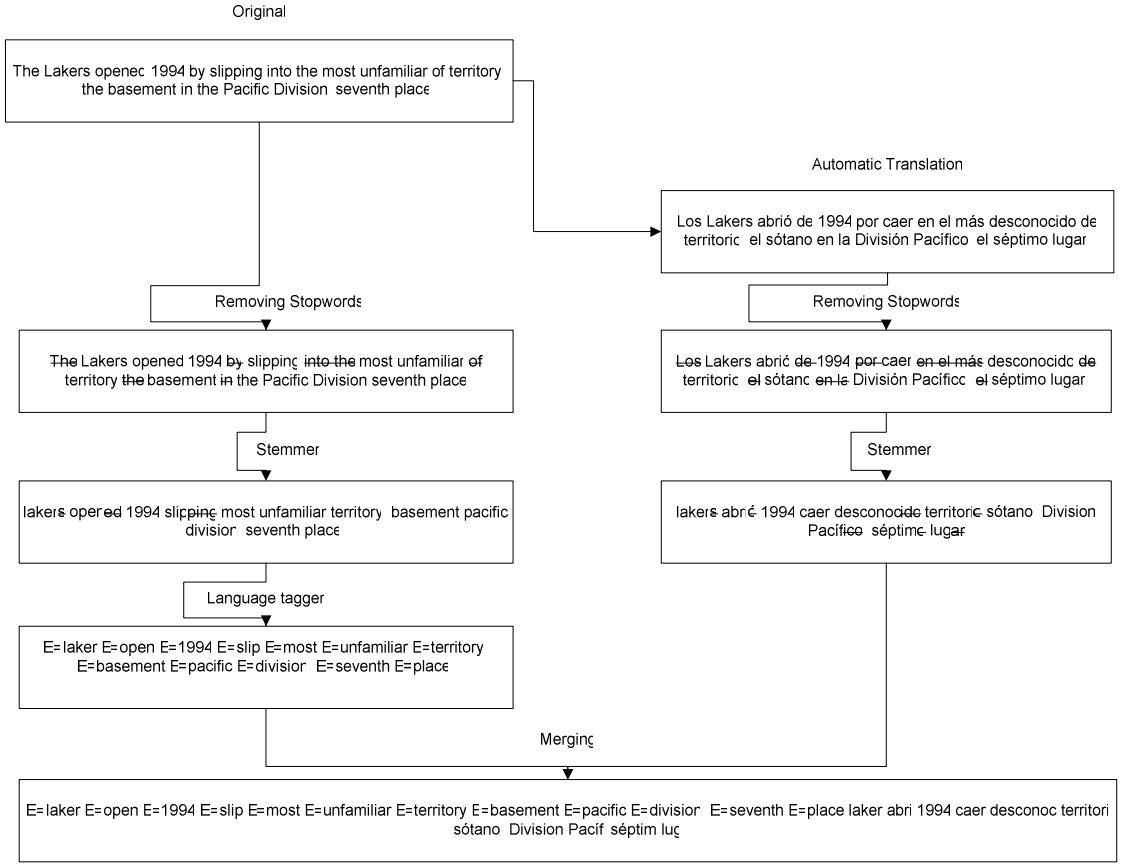

Figure 3 – Steps in the pre-processing phase

ii. **Mining ARs:** This step consists in generating ARs for the terms in the query. We run the Apriori Algorithm over the pre-processed parallel sentences. In order to speed up rule generation, only sentences that contain the query terms are considered. As a result, the support for all rules will be

100%, which means that we can no longer use this metric as an indication of rule usefulness. The output of this phase is a set of ARs for each query term.

iii. **Rule Filtering:** The aim of this step is to keep the rules that most likely map a term in the source language to its translation in the target language. The series of heuristics listed below was developed by observing empirical data. They are applied on the ARs generated for each query term:

a) Discard rules in which the antecedent and the consequent are in the same language. Since we are trying to map terms between languages, these rules are not of interest.

b) Select the AR with the highest confidence. The rule with the highest confidence is more likely to be the correct mapping. Such a rule will be called *M*.

c) Select the ARs that have confidence of at least 80% of *M*.

d) Select ARs with confidence equal to (100 – *M* ± 0.1), as it was observed that words in a language that are normally translated into two (or more) words in another language tend to have complementary confidences.

The application of these heuristics is illustrated in Figure 4.

`civil ⇒ E=war (26.1)`	Discarded – Low confidence (c)
`civil ⇒ E=civilian` `(29.6)`	**Selected** – Complement to 100 (d)
`civil ⇒ guerr (25.6)`	Discarded – Antecedent and consequent in the same language (a)
`civil ⇒ E=civil` `(70.5)`	**Selected** – AR with highest confidence (b)

Figure 4 – Example of filtering association rules for the term "civil". The numbers between brackets are the confidence of the AR

iv. **Query Translation:** Each term in the original query is replaced by all possible translations that remain after the filtering process. The output of this step is the query in the target language.

v. **Query Execution:** The last step if to execute the queries in a search engine. At this stage, the CLIR problem has been reduced to a traditional monolingual query processing. The output is a list of retrieved documents.

It is worth pointing out that the collection used as a basis for the mining of ARs need not be the same collection used for document retrieval. It is possible to extract the ARs from a bilingual corpus and to use a different test collection for document retrieval.

Our approach mines the ARs on demand, according to a lazy strategy as proposed by (Veloso et al., 2007). Thus, we only generate rules for the terms in the query, and as we only consider the sentences in which the query terms appear for rule generation, the number of rules is significantly reduced. On the other hand, this strategy delays query processing. To speed up this process, we could build a cache of ARs, eliminating the need to mine for all the rules at query time.

3 Modifying BM25 to emphasise rare terms

Okapi BM25(Robertson & Walker, 1994) is a ranking function used by search engines to rank documents according to their similarity to a given query. This is a very popular ranking function and it is implemented in many IR systems. In order to improve our IR results, we have implemented modifications to the original BM25 formula, shown in Eq 3.

$$BM25(D,Q) = \sum_{i=1}^{n} \log\left(\frac{N - n(q_i) + 0,5}{n(q_i) + 0,5}\right) * \frac{f(q_i, D) * (k_1 + 1)}{f(q_i, D) + k_1 * \left(1 - b + b * \frac{|AL|}{\mathrm{tf}_{dt}}\right)} \quad \textbf{(3)}$$

where: N is the number of documents in the collection
$n(q_i)$is the number of documents indexed by term q_i

$f(q_i, D)$ is the frequency of term q_i on document D
AL is the number of terms in document D
k_1 and b are parameters, usually chosen as 2.0 and 0.75, respectively

Our modification on BM25 aims at promoting rare terms, i.e. terms that occur in few documents. The modification is divided into two steps. The first step is to reduce the weight of common terms in the collection and it is accomplished by adding a new multiplier to the original function. The weights of the multipliers were defined by observing query results on the LA Times collection and are shown in Eq. 4. We call them "Intermediate Scores" or scoreI.

$$\text{scoreI(D,Q)} = (0,00005\,p_i^{\,4} - 0,019\,p_i^{\,3} + 0.0211\,p_i^{\,2} - 0,0926\,p_i + 1,1697) * BM25\text{(D,Q)} \quad \textbf{(4)}$$

where: $p_i = n(q_i)/N$ is number of documents indexed by the term

The improvement in terms of query results obtained by Eq. 3 is only marginal. It will only achieve significant results when *stop-words* are not removed or in collections with very few documents. As a consequence, a second phase is applied.

The second step aims at promoting rare terms more emphatically. Let m be the average number of occurrences of the terms in the collection. Using m, the number of occurrences of each term $n(q_i)$, and the intermediate scoreI, the modified version of BM25, called *BM25+*, is shown in Eq 5. It is important to notice that the seven conditionals in Eq. 5 are not mutually exclusive. For example, a term appearing in just one of 10,000 documents, would receive all increments in the BM25+ function.

$$BM25+\text{(D,Q)} = \begin{cases} scoreI(D,Q) + 0,05 * \min\left(4, \dfrac{n(q_i)}{m}\right) if \quad n(q_i) < m \\ scoreI(D,Q) + 0,1 \; if \quad n(q_i) < 1000 \\ scoreI(D,Q) + 0,2 \; if \quad n(q_i) < 500 \\ scoreI(D,Q) + 0,3 \; if \quad n(q_i) < 100 \\ scoreI(D,Q) + 0,5 \; if \quad n(q_i) < 50 \\ scoreI(D,Q) + 0,8 \; if \quad n(q_i) < 20 \\ scoreI(D,Q) + 1,5 \; if \quad n(q_i) < 6 \end{cases} \quad \textbf{(5)}$$

4 Experiments

This section describes our experiments submitted to the CLEF-2008 campaign. Section 4.1 details the resources used, and Section 4.2 presents the results.

4.1 Description of Runs and Resources

We worked on the English TEL collection, which contains catalogue data from the British Library. The details of the test collection are described in Table 1. Our aim was to test the feasibility of our proposed approach for using ARs to map concepts between languages. Our bilingual experiments use Spanish queries to retrieve documents in English.

Table 1 - Details of the test collection

Number of unique terms	689,053
Number of documents	1,000,101
Size	195MB

The procedure is the same as described in Section 2. Since our approach needs a sample of parallel documents and the TEL collection does not have parallel documents, we had to translate a sample

of the original documents using Google Translator[1]. The sample size was 25% of the collection (250,025 documents). The sample was taken by picking one in every four documents in sequence.

We removed stop-words according to the lists available from Snowball[2]. The Porter Stemmer (Porter, 1980) was used on the English texts and the Spanish version of the Porter Stemmer (Snowball) was used on the Spanish documents. The IR system we used was Zettair (Zettair), which is a compact and fast search engine developed by RMIT University (Australia) distributed under a BSD-style license. Zettair implements a series of IR metrics for comparing queries and documents. We used Okapi BM25 as some preliminary tests we performed on other data collections showed it achieved the best results.

The time taken to run each query is approximately 45 seconds including the mining of the ARs, rule filtering, query translation and processing by the search engine. The time taken varies according to the number of terms in the query. The longest time is taken by the selection of the sentences that will serve as the basis for the mining process. The tests were performed on a Pentium 4 2.8GHz with 512 Mb of RAM running Windows XP. At this point we were not concerned with performance, thus no caching of rules was implemented.

All runs use stop-word removal. Since our goal was to test our approach on a cross-linguistic setting, our monolingual runs serve only as a baseline. Four runs were submitted:

- UFRGS_BI_SP_EN – uses our proposed method for ARs
- UFRGS_BI_SP_EN2 – uses our proposed method for ARs and BM25+
- UFRGS_MONO_EN1 – baseline monolingual run
- UFRGS_MONO_EN2 – monolingual run using BM25+

4.2 Results

Our results are summarised in Table 2 and Figure 5. Comparing the monolingual and bilingual runs, we notice that the bilingual executions achieve up to 86% of the corresponding monolingual performance in terms of Mean Average Precision (MAP). A T-test showed that the difference in performance between monolingual and bilingual runs is not statistically significant if measured by MAP. This was noticed both for the runs with the original version of BM25 and for the runs with our modified version. Compared to other participants, our bilingual version was ranked in third place. These results indicate that our approach for mapping concepts between languages using ARs is adequate.

When comparing performance in terms of Pr@10, however, our bilingual runs are statistically worse than their monolingual counterparts. This fact can be observed in Figure 5, as the superiority of the monolingual runs is more evident at low recall levels. From recall 0.5 onwards, all runs have very similar results.

Comparing the results obtained by the original BM25 formula and BM25+, we can see that our modification achieves better results. Improvements were noticed for monolingual and bilingual runs both in terms of MAP and PR@10. However, this difference was only considered to be statistically significant for the monolingual run in terms of MAP.

Table 2 - Results in terms of MAP and Pr@10

Run	Mean Average Precision	Precision at 10
UFRGS_BI_SP_EN1	0.2151	0.3120
UFRGS_BI_SP_EN2	0.2315	0.3560
UFRGS_MONO_EN1	0.2493	0.4120
UFRGS_MONO_EN2	0.2777	0.4440

[1] http://www.google.com/translate_t

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

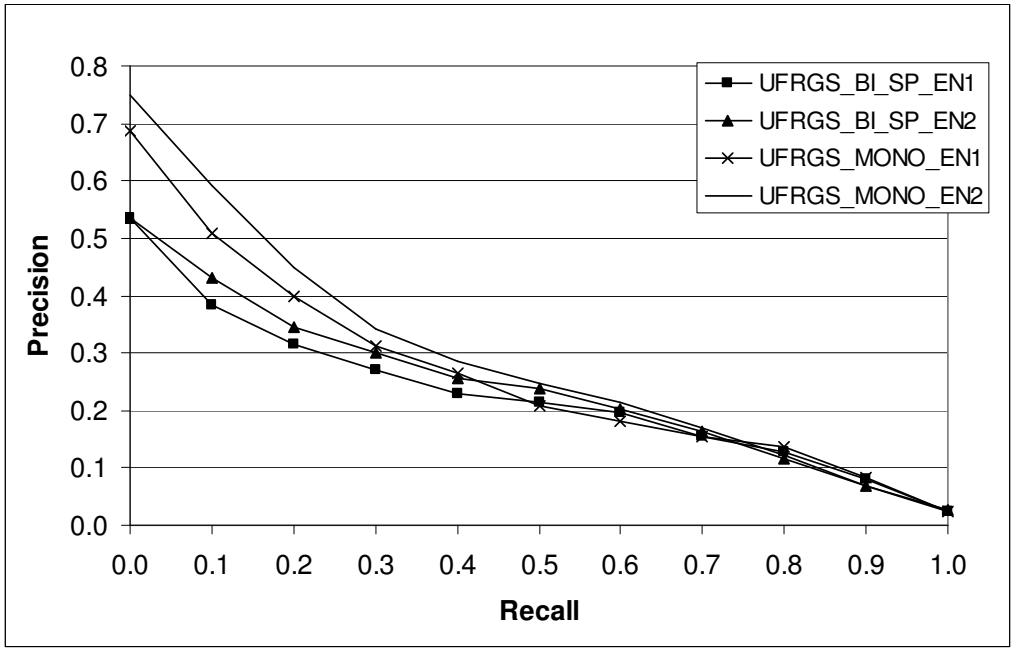

Figure 5 - Recall-precision curves

5 Conclusions

This paper reported on monolingual and bilingual ad-hoc information retrieval experiments that we have performed for the TEL task. Our aim was to validate our proposal of using algorithms for mining association rules for CLIR. The results of the experiments show that our bilingual runs achieve 86% of the performance of the monolingual runs. More importantly, is that the difference in MAP is not statistically significant, which shows our approach is feasible. Since we used automatic translation to generate a sample of parallel documents and it is widely known that these algorithms are far from perfect, it is possible that our results would be better if had translation was used. This fact still needs further investigation.

We have also tested a modification we proposed over Okapi BM25 to increase the weight of rare terms. The results show that the modified version, which we called BM25+, achieves better results.

The experiments reported here provided encouraging results. However, there are still a number of open issues that will be explored as future work; they include: assessing the impact of the size of the sample used for translation in the results; comparing results obtained using an automatic translator to generate a parallel collection against the results obtained using a higher quality (hand-translated) parallel corpus.

Acknowledgements

The authors would like to thank Marcos Gonçalves for his valuable advice. This work was partially supported by CNPq Universal 484585/2007-0. Andre Geraldo is funded by a studentship from CNPq.

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. In *Proc. of the ACM SIGMOD Conference on Management of Data*. Washington, D.C.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th VLDB Conference* (pp. 487-499). Santiago, Chile.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, 41*(6), 1-13.

Hipp, J., & Güntzer, U. (2002). Is pushing constraints deeply into the mining algorithms really what we want?: an alternative approach for association rule mining. *ACM SIGKDD Explorations Newsletter, 4*(1), 50-55.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program, 14*(3), 130-137.

Robertson, S., & Walker, S. (1994). Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC)*. Gaithesburg, Maryland.

Snowball.*Spanish Stemmer*. Retrieved 08-Aug-2008, from http://snowball.tartarus.org/algorithms/spanish/stemmer.html

Veloso, A., Meira Jr., W., Gonçalves, M. A., & Zaki, M. (2007). Multi-label Lazy Associative Classification. In *PKDD - LNAI 4702* (Vol. 4702, pp. 605-612): Springer.

Zettair. Retrieved 11/06/07, 2007, from www.seg.rmit.edu.au/zettair/