

A Novel Implementation of the FITE-TRT Translation Method

Aki Lopenen¹, Ari Pirkola¹, Kalervo Järvelin¹ and Heikki Keskustalo¹

¹ Department of Information Studies, University of Tampere, Finland.
{Aki.Lopenen, Kalervo.Jarvelin, Heikki.Keskustalo}@uta.fi, pirkola@cc.jyu.fi

Abstract. Cross-language Information Retrieval requires good methods for translating cross-lingual spelling variants which are not covered by the available dictionary resources. FITE-TRT is an established method employing frequency-based identification of translation equivalents received from transformation rule based translation. This study further develops and evaluates the FITE-TRT method. The paper contributes on four areas. First, an efficient implementation for the FITE-TRT method is discussed. Secondly, a novel iterative FITE-TRT translation approach is developed in order to further improve the effectiveness of the method. Thirdly, the effectiveness of FITE-TRT is assessed in three classes of source-target word similarity. FITE-TRT was found to be very strong in the class of the most similar source and target words and only becomes unsuccessful when the words were dissimilar. Fourthly, in comparison to n-gram and s-gram matching methods, FITE-TRT is shown consistently stronger. All in all, FITE-TRT clearly outperforms the fuzzy string matching methods under comparable conditions. Therefore it is the method of choice for the identification of translation equivalents of cross-lingual spelling variants when the requirements for the result quality are high.

Keywords: Approximate string matching, cross-language information retrieval, cross-lingual spelling variants, fuzzy matching, out-of-vocabulary words, transformation rules.

1 Introduction

Frequency-based identification of translation equivalents received from transformation rule based translation (FITE-TRT) is a method which addresses the out-of-vocabulary (OOV) problem in cross-language information retrieval (CLIR) by providing an effective method for translating spelling variants [5]. FITE-TRT consists of two consecutive methods: the TRT and the FITE. TRT receives a *keyword* in source language and outputs a list of *translation candidates* in a target language. FITE processes these candidates and outputs a result - the proposed translation or a flag indicating that a translation cannot be identified.

The TRT phase, originally presented in [6], generates a number of translation candidates by applying suitable *transformation rules* to a given keyword. A transformation rule contains source language characters that are transformed into the

target language characters given in the rule, and their context characters. A rule also has two numerical factors: *confidence factor* and *frequency*, which determine the importance of a rule. Frequency refers to the number of occurrences of the rule in the dictionary data that was used in rule generation and confidence factor is defined as the frequency of a correct rule application divided by the number of source words where the source substring of the rule occurs.

The FITE phase scans through a list of candidates generated by TRT and gives either exactly one translation or an empty output (no translation cases) in contrast to approximate string matching [4][7] where the source word always matches some target words (thus there are no 'no translation' cases). The FITE phase has three conditions and if those are fulfilled then a translation can be given. The first one is the *beta condition* which checks that the frequency of the candidate with the highest frequency value in a target language is more than a predefined *beta*-value (β) times the frequency of the second best candidate. If the first and second best candidates do not fulfil the beta-condition requirements, the second best candidate is compared with the third best. If the comparison meets the beta condition, then the first candidate is selected, because the most common among similar candidates is the most probable candidate for translation. An example of beta condition is presented below with following words and their frequency values:

- lucille 20,000
- lucile 5000
- lusille 200

If $\beta=2$, then the first comparison between the words 'lucille' and 'lucile' satisfies the condition ($20,000 > 2*5000$) and thus the 'lucille' is qualified. If $\beta=10$, then the comparison between first two words fails ($20,000 \leq 10*5000$). Next the frequency values of words 'lucile' and 'lusille' are compared and now the condition is satisfied ($5000 > 10*200$) and again the 'lucille' can be qualified. Both stages fail if $\beta=25$.

The second condition checks that the relation between the frequency of a candidate in a target language and the frequency of a source word in a source language is valid. The frequencies are normalized using predefined parameter *alpha*, thus the condition is called *alpha-condition*. FITE takes the frequency information from word frequency lists specifically constructed for this purpose. The third condition (the length factor) checks that the length difference between the key and candidate is reasonable.

The present paper focuses on four issues. First, an efficient implementation for the FITE-TRT method is developed for the first time. Secondly, novel iterative FITE-TRT translation strategies are studied in order to further improve the effectiveness and efficiency of the method. The idea is to translate the source words stepwise, gradually relaxing the FITE-TRT parameters. By first applying stringent criteria, the number of target word candidates remains small. If a translation is not identified for some word, then more relaxed criteria are employed – and more candidates generated. Thirdly, the effectiveness of FITE-TRT is assessed in three classes of source-target word similarity. In this paper it is shown that FITE-TRT handles well translations of source words that are at least moderately similar to their translations – better than known alternatives. Fourthly, FITE-TRT is compared to n-gram and s-gram matching [1] methods in a large-scale test which demonstrates that FITE-TRT is highly competitive. Finally, an analysis on how many fuzzy translations are required to

achieve the recall of FITE-TRT is performed. All in all, the tests are to show that FITE-TRT clearly outperforms the fuzzy matching methods under comparable conditions and can be implemented efficiently. Therefore it is the method of choice for the identification of translation equivalents of cross-lingual spelling variants when the requirements for the result quality are high.

The paper is organized as follows: First, an efficient implementation for the FITE-TRT method is considered in Section 2. The novel iterative FITE-TRT translation strategies are presented in Section 3. The effectiveness of FITE-TRT method is evaluated and also compared to *n-gram* and *s-gram* matching in Section 4, followed by conclusions in Section 5.

2 New Features of FITE-TRT

In this section an example keyword along with few suitable TRT-rules are used to illustrate the FITE-TRT process. The keyword is Spanish word “aditivo” which has an English translation “additive”. The TRT-rules utilized are {adi addi b 6 42.86}, {ti tai c 2 0.08}, {tivo t e 1 0.69} and {vo ve e 123 62.44}, where the first character string is the source language substring (which is replaced in the key), the second character string is the substring for target language (which replaces the source language substring in the key) and the third separate character indicates the position of the rule: *b* means that the rule targets the beginning of the key, *c* means center and *e* means that the end of the key is targeted. The integer represents the frequency value of the key and the decimal number is rule’s confidence factor.

2.1 New Implementation

The basic FITE-TRT implementation [5] was effective but not very efficient as such. It introduced the *windowing of rules* by their confidence factor and frequency as a means to reduce the number of generated translation candidates, but it can still create vast numbers of candidates. Resources can be saved when the consecutive TRT and FITE processes are merged into a joint process (Figure 1).

Retrieving frequency information from external data storage for each generated candidate is very ineffective. In the basic implementation each candidate requires one query operation in target language frequency data. For example, using all Spanish-English TRT-rules defined in [6] for the Spanish the keyword “aditivo” yields 16,400 translation candidates and as many frequency data queries.

Optimization is achieved by trimming the number of frequency data queries in two ways. First the number of generated candidates is reduced. Direct limit to the number of generated candidates will not work, because the correct candidate can be any of the created. While windowing the ruleset is quite an effective pruning method, efficiency is a bit arbitrary, because still unknown numbers of rules become selected: long keys can have plenty of suitable rules while short keys can only have a few if any.

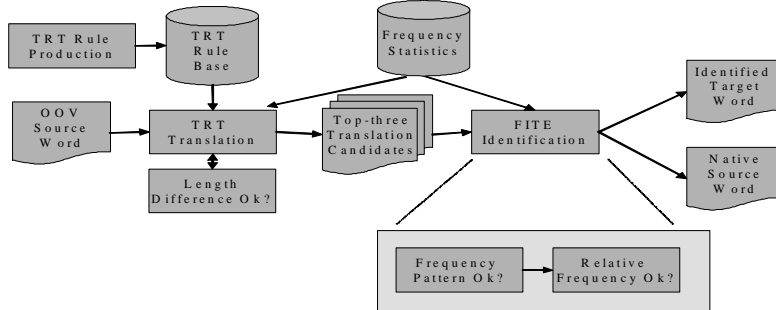


Fig. 1. The merged FITE-TRT process.

A method for *weighting of the rules* is adopted to ensure a fixed upper limit for the number of applied rules. Suitable rules are ordered into descending order by their *weight* which for single rule r is calculated using formula (1), where f is the frequency of the rule, cf its confidence factor, af is the sum frequency of all the rules, and pf is the sum frequency of all the rules affecting the same source language characters as r . Now the number of generated candidates is more controllable than with windowing and presumably the quality of utilized rules is better. Weighting of the rules adds the *rule number* parameter, which defines the number of best rules to be selected from all rules compatible with a given key. However, a key which has several rules fitting the same part of the key and has several of such parts will still produce lots of candidates.

$$weight(rule) = \frac{f * cf}{af * pf} \quad (1)$$

Merging some of FITE's functionality into candidate generation also reduces the queries into the data storage structures. As described above, the FITE method has three conditions that a candidate must fulfil to be accepted as a translation. *Length factor* is the third condition, but there is no valid reason for not to utilize it while generating candidates. If a generated candidate does not meet the length criteria, then target language frequency for it is not retrieved and processing continues to the next candidate. In the optimized implementation the candidates are generated recursively in preorder. The main root of the recursive key generation tree is the source key itself and it has as many child nodes as there are possible single rule adaptations. Adaptations for the key's rule slots are done from left to right. Further adaptations for a generated candidate are done to the part right from previous adaptations window. The maximum depth of the recursion tree is the highest number of rules that can be reconciled with the keyword. An example of the recursive candidate tree for the key and rules given in the beginning of this section is presented in Figure 2. The bold section of each candidate string represents the part of the candidate which has been influenced by a rule. When traversed in pre-order, the candidates generated are in following order: aditivo, additivo, additaivo, additaive, adit, additive, aditaivo, aditaive, adit, aditive. (In this example the rules have not been windowed or weighted.)

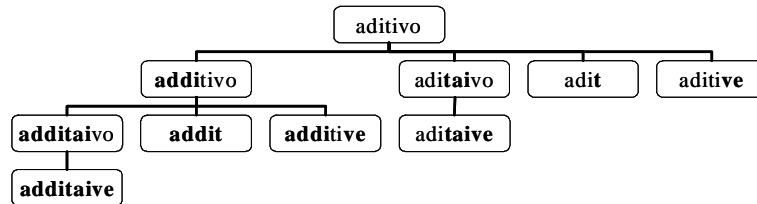


Fig. 2. An example of the recursive candidate generation tree.

Only three candidates bearing the highest frequency values from target language frequency data are kept in memory instead of storing all the generated candidates. When all the candidates have been created, the two remaining FITE conditions are applied into the top-three list and the topmost candidate in the list is accepted or rejected.

2.2 Division Between Short and Long Words

When experiments with FITE-TRT were made, it was noticed that the same combination of parameters was not necessarily optimal for words with significant difference in their lengths. Better results were achieved when shorter words were processed with a different parameter combination than the longer ones. For all the languages used, the length of 7 letters was found to be as the best dividing value. Every keyword shorter than, or exactly, seven letters in length was handled as a short word and each longer keyword as a long one. This word length factor is genuine because longer words have more possible positions for rule transformations than shorter ones.

2.3 Frequency Data

FITE needs an associated program that collects Web data and constructs *word frequency lists* for words of a given language. The list can become very large. In particular the English list covers extensively the English lexicon but it is also littered with misspelled words and some foreign language words (because language filtering is not 100% accurate). Sometimes a candidate obtained in TRT matches with a litter word contained in the frequency list. By further filtering the lists the effectiveness of FITE-TRT is expected to improve. The number of unique words contained in the frequency lists gathered for the experiments is as follows:

- English: 1,643,000 words
- French: 771,000 words
- German: 1,422,000 words
- Spanish: 957,000 words

2.4 Performance Gains

The novel implementation was programmed in Java which enabled portability to several platforms as well as good modularity for different kinds of applications. When the frequency data and the TRT-rules are loaded into memory prior to translations, the typical resolution time for a key was around 25-35 milliseconds (using Windows XP in a PC with Intel DualCore2 6300 processor with 2 GB's of RAM and 1600 MB's allocated to Java Runtime Environment).

While this resolution time is lengthy for real time use, the keys to be translated are essential in (short) queries and therefore it makes sense to pay the effort. Also when compared to previous way of implementing the FITE-TRT, this novel implementation is justified. Implementing the FITE-TRT manually could take days, even weeks for a single key. A brute-force method where TRT- and FITE-phases are applied successively, can easily take hours for a single long key.

3 Iterative FITE-TRT

The original FITE-TRT processing utilizes a single combination of parameters, which are alpha- and beta-values, length criteria for a key and candidates, and the number of best rules employed. This language dependent combination is trained to produce the highest recall and precision values and therefore the best possible overall result for FITE-TRT. Some number of words will be left untranslated, because there is no single parameter combination that could reliably translate every valid word. Therefore an iterative method is presented here as an extension to the original method.

The iterative FITE-TRT will apply different parameter combinations consecutively until the candidate is accepted or until it becomes certain enough, that the key cannot be translated. Parameter combinations are evaluated with the numbers of correct and incorrect translations, and the number of untranslated words. During iteration incorrect translations are avoided by translating carefully in each level of iteration. Each parameter combination perfectly translates some words while it can also translate other words incorrectly. Therefore effective iteration requires such a sequence of parameter combinations, that most of the words are translated as correctly as possible.

3.1 Strategies for the Identification of Iteration Parameters

In order to identify the best sequence of parameter combinations three strategies are presented, because it was not obvious, at the outset, how a good iteration could be formed. All strategies have the risk of overfitting to the training data and all strategies are also *static* in a way that they have to be tuned each time the TRT-rules or language frequency data changes. In the next section the training of these strategies is evaluated.

The original FITE-TRT used simply the best single parameter combination that resulted in as high recall and precision values as possible. This strategy is used as a *baseline* in evaluating the iterative strategies.

The first strategy is a *manual* search for a suitable sequence of parameter combinations. This translates words in consecutive steps, or in iteration. The words that do not translate in the first step are handled in the second step, and so on. In each step the aim is to achieve as high a recall as possible while at the same time maintaining 100% precision. If 100% precision is not achieved the parameters of the steps are changed, or the process is restarted from a previous step whose parameters are changed in such way that 100%, or as high as possible, precision is still achieved (“back-tracking”). The process continues until words cannot be translated any more. The difficult words accumulate to the end of the process, and these words remain untranslatable. This strategy’s main problem is the huge manual effort required because of the back-tracking.

The second strategy is called the *best&iterative* strategy. It tries to improve the baseline strategy by applying it in subsequential steps until all translatable words are translated. Here the combination which yields the highest recall value with high precision is selected. The untranslated training words are the new keywords for the next step, where all reasonable parameter combinations are again used to translate the new keywords and - just like in the baseline strategy - the best combination is selected. This is continued until no more new translations with reasonable results can be made. The choice between high valued combinations is done manually, since the cases can be fairly different. This strategy takes less time to prepare than the first one.

The third strategy - the *min-error* strategy - is to select the combination that yields the minimal number of incorrect translations. In case of a tie, the combination with highest recall is selected. This strategy results in a long chain of combinations and it takes a lot of time to be found, but is fully automatic. Untranslated keywords are mined again as in the second strategy. The “back-tracking” step of the first strategy is not made, since it can result in explosive growth of recursion.

3.2 Iteration Tests and Results

The iterative translation strategies were developed using three *training word sets*, one set for each language pair. These word lists were received from a researcher who has investigated cross-lingual spelling variant matching. The training lists contained 463 (French-English), 468 (German-English), and 546 (Spanish-English) word pairs.

In the final iterative experiments, the effectiveness of iterative translation was evaluated using three sets of test words, again one set for each language pair. The test words were extracted from the Multilingual Glossary of Medical Terms by Heymans Institute of Pharmacology, University of Gent [3]. All entries where the entry words in the considered languages were single words were extracted from the Glossary. Because FITE-TRT is intended to translate similar words, for the final test word lists only word pairs whose words were sufficiently similar with one another were selected. As a threshold the similarity value of $LCS/LW=0.60$ was used (LCS/LW is a longest common subsequence based similarity measure, see Section 4.1.). In addition, the words that were used in the training experiments were removed from the final

lists. The final lists contained 1013 (French-English), 1014 (German-English), and 1009 (Spanish-English) unique word pairs.

Obtaining training and test word sets from dictionaries doesn't create a conflict with the objective to create a reliable method for spelling variant OOV words. For technical terms, the transformation rules of the spelling variant word translations between two languages are similar whether the words are in or out of vocabulary, i.e. orthographical conventions are constructed with quite homogenous linguistic rules.

Because the manual strategy takes huge amounts of time to prepare, it was only done for the Spanish-English language pair. However the manual iteration sequence was applied to other language pairs to make a point that universal iteration is not likely to exist, or at least it is not easy to find.

The results for all strategies and language pairs are presented in Table 1. Column "best&iter" shows the results for the second strategy. The columns "manual" and "min error" stand for the other strategies, where the former is the manually searched iteration sequence for Spanish to English translation employed on all language pairs and the latter is the sequence generated with fully automatic parameter mining, which is trained to maximise precision at the expense of recall.

Table 1. Recall and precision values for all strategies and baseline translation in all language pairs.

| | | baseline | best&iter | min error | manual |
|----------------|---------------|----------|-----------|-----------|--------|
| Spa-Eng | Recall (%) | 80.2 | 82.2 | 82.0 | 86.4 |
| | Precision (%) | 86.6 | 86.0 | 83.1 | 88.1 |
| Ger-Eng | Recall (%) | 73.3 | 73.8 | 74.3 | 73.5 |
| | Precision (%) | 85.3 | 85.3 | 85.3 | 81.7 |
| Fre-Eng | Recall (%) | 75.5 | 77.2 | 78.3 | 77.1 |
| | Precision (%) | 82.9 | 82.0 | 82.8 | 81.2 |

The baseline values were as expected. Spanish yielded the highest recall and German the lowest. This was the case for all other strategies as well. The German rules and frequency data contained more characters and created more diversity to rules and source words. Precision values were quite close to each other cross-lingually.

The thoroughness of the baseline was proved by the best&iterative strategy, since the improvements in recall were marginal (Spanish gained the most by two percent units while German only gained half a percent unit) and strategy's precision either remained or got slightly worse. The baseline strategy already translated most words and left only difficult words without translation. Translating a wide variety of problem words without overfitting parameters to those specific words is a difficult task.

The minimal error -strategy tried to beat the baseline strategy by translating words in smaller groups while avoiding incorrect translations. Training this strategy resulted in long iterations: from 25 to around 30 levels. Recall improved slightly from baseline for all three languages, but precision either remained or got slightly worse. The manually constructed iteration for Spanish to English translation outperformed the baseline. Recall was 86.4% and precision 88.1% against the baseline's 80.2% and

86.6%. The manual strategy also beat other strategies in Spanish to English -pair mainly because it utilized backtracking steps when combining the iteration parameter sequence.

4 FITE-TRT vs. Fuzzy Matching

The effectiveness of FITE-TRT is compared to that of fuzzy matching (n-gram and skip-gram matching). N-grams have been found effective for fuzzy matching in IR [4], [7] and its generalization, the skip-grams (s-gram for short), consistently outperformed n-grams in the identification of translation equivalents of cross-lingual spelling variants [1]. Here the term n-gram refers to di-grams formed of consecutive characters of words. The s-gram fuzzy matching technique constructs di-grams both of consecutive and non-consecutive characters of words [1]. The generated di-grams are put into comparable categories based on the number of skipped characters as di-grams are constructed. The character combination index (CCI) indicates the number of skipped characters as well as the comparable categories. Here the CCI= $\{\{0\}, \{1, 2\}\}$ was used. This means that di-grams formed of consecutive characters form one comparable category and di-grams with one and two skipped characters the other. S-grams formed in this way consistently outperformed conventional di-grams in [1].

For example, with string $S = abcd$ and $CCI = \{\{0\}, \{1, 2\}\}$, two digrams sets are formed, namely $DS\{0\}(S) = \{ab, bc, cd\}$ (by zero skipping) and $DS\{1, 2\}(S) = \{ac, ad, bd\}$ (by skipping both one and two characters).

The strength of n-/s-grams is that they do not need frequency lists and may be operated in the target database index, thus they are more efficient than FITE-TRT.

4.1 The Experiments

The dependence of FITE-TRT, n-gram and s-gram effectiveness on the similarity degree of the source and target words was considered in these experiments. The test words were extracted from the Multilingual Glossary for Art Librarians [2] which is well suited for this test because it contains both similar and dissimilar words. FITE-TRT translates individual words and only such entries from the Glossary were extracted where English, French, German, and Spanish entry words appear as individual words. This is no loss of generality as OOV phrases can be TRT'ed word by word.

Three test word sets were constructed, one for each language pair. All the word sets contained the same word pairs ($n=123$) albeit in different languages. All the 3 x 123 word pairs were looked at, and if the original target word was not the most similar one to a source word, it was removed and a new target word, the most similar one, was taken from the dictionary as the equivalent of a source word.

A simple measure of longest common subsequence divided by the length of the longer word of the word pair (LCS/LW) was used as a similarity measure. The closer to 1.00 LCS/LW is, the more similar the words are: $LCS/LW = 1.00$ refers to identical words. As an example, for the Spanish word *suplemento* $LCS/LW = 9/10 = 0.90$ w.r.t. the English equivalent *supplement*. For the native Spanish word *vinculante*

LCS/LW = 0.30 w.r.t. its equivalent *binding*. The test words were classified into three categories based on their LCS/LW values: (1) 0.80-100, (2) 0.60-0.79, and (3) 0.00-0.59. The FITE-TRT effectiveness w.r.t. n-gram and s-gram matching effectiveness is analyzed within these categories.

The test words in French, German, and Spanish were translated into English by FITE-TRT and were matched against the English word list using n-grams and s-grams. The English word list used in fuzzy matching was the same list as used by FITE-TRT, except that it only contained words without frequency figures. In the first experiment the effectiveness of FITE-TRT vs. fuzzy matching was considered from the viewpoint of FITE-TRT in that only the highest ranked word in the ranked result list of fuzzy matching was considered. The highest ranked word yielded by fuzzy matching was either a correct or incorrect translation for a source word. The matching results were used to compute recall / precision for n-grams and s-grams.

FITE-TRT was run without iteration using two parameter combinations (PC): PC1: 10 best rules (by weight), $\alpha = 2$, and $\beta = 2$; and PC2: 3 best rules, $\alpha = 10$, and $\beta = 10$. In both cases the length difference between the source word and target word was set at 0-3 characters for long words and 0-2 characters for short words.

The recall of fuzzy matching can be improved by increasing the number of target words that are considered in the result list. The second experiment determined what is the minimum number of target words (MNW) that need to be collected to obtain a recall higher than the one obtained by FITE-TRT. For example, MNW=7 means that on average fuzzy matching achieves higher recall than FITE-TRT when the best seven words are considered, and with six words recall in fuzzy matching remains lower than in FITE-TRT. In this experiment the performance of s-grams (which consistently outperformed n-grams) was compared against the performance of FITE-TRT with parameter combination 2 (the worse of two parameter combinations regarding recall). In this test, the recall and precision of fuzzy matching are not equal since more than one target words per source word are considered. In this experiment s-gram matching is compared with FITE-TRT on the similarity level of LCS/LW=0.80-100.

4.2 Findings

Table 2 reports the results of the experiment, described in Section 4.1, where FITE-TRT's performance was compared against the performance of fuzzy matching. On each similarity level the number of test words varies by the language pair. Note that for each language pair the total number of test words was $n=123$. The figures suggest that French and English words are more similar to each other than German-English words and Spanish-English words since on the level of LCS/LW=0.80-100 the number of test words is higher for French-English than for the other pairs. The recall and precision values are equal for s- and n-grams because, in this test, the grams always produce one translation candidate which is either correct or incorrect.

Table 2. FITE-TRT, s-gram and n-gram effectiveness.

| Translation / matching method | LCS/WL > 0.79 | | | 0.6<LCS/WL<0.8 | | | 0.0<LCS/WL<0.6 | | | Overall |
|-------------------------------|---------------|------|------|----------------|------|------|----------------|-----|------|------------|
| | N | R% | P% | N | R% | P% | N | R% | P% | F-measure% |
| FITE-TRT/PC1 | | | | | | | | | | |
| Fre-Eng | 84 | 78.6 | 88.0 | 23 | 34.8 | 44.4 | 16 | 0 | 0 | 64.0 |
| Ger-Eng | 52 | 71.2 | 78.7 | 26 | 26.9 | 38.9 | 45 | 2.2 | 10.0 | 40.4 |
| Spa-Eng | 53 | 79.2 | 87.5 | 45 | 44.4 | 58.8 | 25 | 8.0 | 14.3 | 56.6 |
| FITE-TRT/PC2 | | | | | | | | | | |
| Fre-Eng | 84 | 66.7 | 96.6 | 23 | 30.4 | 58.3 | 16 | 0 | 0 | 61.5 |
| Ger-Eng | 52 | 69.2 | 90.0 | 26 | 23.1 | 60.0 | 45 | 2.2 | 20.0 | 43.6 |
| Spa-Eng | 53 | 62.3 | 91.7 | 45 | 22.2 | 43.5 | 25 | 4.0 | 16.7 | 44.5 |
| S-gram | | | | | | | | | | |
| Fre-Eng | 84 | 65.5 | 65.5 | 23 | 4.3 | 4.3 | 16 | 0 | 0 | 45.5 |
| Ger-Eng | 52 | 44.2 | 44.2 | 26 | 0 | 0 | 45 | 0 | 0 | 18.7 |
| Spa-Eng | 53 | 32.1 | 32.1 | 45 | 0 | 0 | 25 | 0 | 0 | 13.8 |
| N-gram | | | | | | | | | | |
| Fre-Eng | 84 | 60.7 | 60.7 | 23 | 0 | 0 | 16 | 0 | 0 | 41.5 |
| Ger-Eng | 52 | 42.3 | 42.3 | 26 | 0 | 0 | 45 | 0 | 0 | 17.9 |
| Spa-Eng | 53 | 32.1 | 32.1 | 45 | 0 | 0 | 25 | 0 | 0 | 13.8 |

It can be seen in Table 2 that FITE-TRT performs much better than fuzzy matching on similarity level of LCS/LW > 0.79. FITE-TRT/PC1 achieves 71.2%- 78.6% recall (R) and 78.7%-88.0% precision (P). For FITE-TRT/PC2 recall is lower but precision is higher than for FITE-TRT/PC1. S-grams outperform n-grams. Regarding recall, only French-English/s-grams is competitive with FITE-TRT achieving a 65.5% recall. However, FITE-TRT/PC1's precision value (88.0%) is clearly higher than s-grams' value (65.5%). Other language pairs cannot really meet up with French-English.

Importantly, on the similarity level of LCS/LW=0.60-0.79 fuzzy matching loses its ability to identify target language equivalents whereas FITE-TRT performs fairly well. FITE-TRT loses its ability to translate words only on the similarity level of LCS/LW=0.00-0.59.

The overall column reports the F-measure for the entire word sets and for each method. In calculating the F-measure, recall and precision were held equally important. The overall tendencies already discussed in the three word similarity classes are confirmed.

Table 3 shows the results of the experiment where MNW was determined for s-grams. As shown, for French MNW is 2, for German 6, and for Spanish 3. In other words, to obtain the recall level of the worst case FITE-TRT two (Fre-Eng), six (Ger-Eng), and three (Spa-Eng) words have to be scanned in the result list of skip-gram matching. In each s-gram case precision is much lower than FITE-TRT's precision. French-English fuzzy matching comes closest to FITE-TRT in MNW and precision but remains still clearly inferior.

Table 3. MNW for s-grams (LCS/WL = 0.80-1.00).

| Translation / matching method | MNW | Recall % | Precision % |
|-------------------------------|-----|----------|-------------|
| FITE-TRT/PC2 | | | |
| Fre-Eng (n=84) | - | 66.7 | 96.6 |
| Ger-Eng (n=52) | - | 69.2 | 90.0 |
| Spa-Eng (n=53) | - | 62.3 | 91.7 |
| S-gram | | | |
| Fre-Eng (n=84) | 2 | 69.0 | 50.0 |
| Ger-Eng (n=52) | 6 | 71.2 | 16.7 |
| Spa-Eng (n=53) | 3 | 69.8 | 33.3 |

5 Conclusions

In this study, an effective FITE-TRT translation system was implemented and evaluated. The merge of FITE and TRT made the FITE-TRT process straightforward and more efficient as an actual computer implementation. The system was used in iterative translation experiments and in FITE-TRT vs. fuzzy matching translation / matching experiments. The results of iterative translation experiments serve as a basis to further develop iterative FITE-TRT. The manually constructed iteration for Spanish to English translation proved that an iterative method can be better than the baseline method. Still, the costs of building a reliable iteration approach seem prohibitive compared to its benefits. It was also shown that FITE-TRT performs considerably better than n- or s-gram matching which further lose on effectiveness and efficiency if translation recall is emphasized (i.e. a low precision tolerated). Based on these results the FITE-TRT technique is recommended for handling untranslatable technical terms in cross-language retrieval and other information systems where automatic translation is part of the system and when the requirements for the result quality are high.

References

1. Keskustalo, H., Pirkola, A., Visala, K., Leppänen, E. & Järvelin, K. Non-adjacent digrams improve matching of cross-lingual spelling variants. SPIRE'03 Conf Manaus (2003) 252 - 265.
2. Multilingual Glossary for Art Librarians (<http://www.ifla.org/VII/s30/pub/mg1.htm>)
3. Multilingual Glossary of Medical Terms by Heymans Institute of Pharmacology, University of Gent (<http://users.ugent.be/~rvdstich/eugloss/welcome.html>)
4. Navarro, G. A guided tour to approximate string matching. ACM Computing Surveys 33(1) (2001) 31-88.
5. Pirkola, A., Toivonen, J., Keskustalo, H. & Järvelin, K. FITE-TRT: A high quality translation technique for OOV words. Proceedings of the 21st Annual ACM Symposium on Applied Computing. Dijon France April 23-27 (2006) 1043 – 1049.
6. Pirkola, A., Toivonen, J., Keskustalo, H., Visala, K. & Järvelin, K. Fuzzy translation of cross-lingual spelling variants. Proc. 26th ACM SIGIR Conf. Toronto (2003) 345 - 352.
7. Zobel, J. & Dart, P. Phonetic string matching: lessons from information retrieval. Proc. 19th ACM SIGIR Conf. Zurich (1996) 166-172.