**Preprint from:** Loponen, A. & Järvelin, K. (2010). A Dictionary- and Corpus-Independent Statistical Lemmatiser for Information Retrieval in Low-Resource Languages. In: Agosti, M. & al. (Eds.), Multilingual and Multimodal Information Access Evaluation, Proceedings of the International Conference in the Cross-Language Evaluation Forum, CLEF 2010, Padua, Italy, September 2010. Heidelberg: Springer, pp. 3-14. Full text at: http://www.springerlink.com/content/33211h5l23885k86/fulltext.pdf

# A Dictionary- and Corpus-Independent Statistical Lemmatizer for Information Retrieval in Low Resource Languages

Aki Loponen & Kalervo Järvelin

University of Tampere, Finland
{fisrtsname.lastname}@uta.fi

**Preprint from:** *Proc. CLEF 2010 Conference, September 2010, Padova, Italy. Final version in LNCS (Lecture Notes in Computer Science), Springer, 2010*

**Abstract.** We present a dictionary- and corpus-independent statistical lemmatizer StaLe that deals with the out-of-vocabulary (OOV) problem of dictionary-based lemmatization by generating candidate lemmas for any inflected word forms. StaLe can be applied with little effort to languages lacking linguistic resources. We show the performance of StaLe both in lemmatization tasks alone and as a component in an IR system using several datasets and query types in four high resource languages. StaLe is competitive, reaching 88-108 % of gold standard performance of a commercial lemmatizer in IR experiments. Despite competitive performance, it is compact, efficient and fast to apply to new languages.

**Keywords.** Information Retrieval, Lemmatization, Out-of-Vocabulary Words, Transformation Rules

## 1. Introduction

Word inflection is a significant problem in information retrieval (IR). In monolingual IR, query word inflection causes mismatch problems with the database index. Likewise, in cross-lingual IR (CLIR) inflected query words, tokens, cannot be found as translation dictionary headwords. These challenges plague morphologically complex languages but disturb retrieval also in simpler ones.

The problems of inflection have been addressed using both stemming (e.g. [4], [8], [12], [13]) and lemmatization (e.g. [6], [7]). The potential benefits of lemmatization over stemming, especially in morphologically complex languages, are increased precision due to less ambiguity in all text-based IR and, in CLIR, the support to accurate token translation by directly matching dictionary headwords. Lemmatizers

traditionally use morphological rules and dictionaries [7]. Dictionary-based methods are however powerless when encountering out-of-vocabulary (OOV) words. OOV words are significant in all IR, because they often are specific in representing the information needs behind short queries [6, p. 31]. The existing contemporary approaches to dictionary independent lemmatization are based on supervised (e.g. [10], [17]) and unsupervised (e.g. [19]) machine learning techniques. Supervised techniques need a training corpus involving a large set of morphologically analyzed tokens. The disadvantage of this method is that the preparation of the training corpus is time-consuming, in particular when the tokens are tagged manually. The main limitation of unsupervised learning techniques is their general and corpus dependent nature that decreases their performance in specific tasks such as lemmatization.

In this paper, we present a statistical corpus- and dictionary-independent lemmatizer for IR, *StaLe*, that effectively deals with all words encountered in texts and queries, OOV words included, and that does not need any tagged corpora. The lemmatizer is based on statistical rules created from a relatively short corpus of token-lemma pairs where each token is an inflected or derived form of a lemma. The method is an extension of the transliteration rule based translation method for cross-language OOV words shown very effective by Pirkola and colleagues [16]. StaLe generates candidate lemmas for any word form. The method can be applied with little effort to languages lacking linguistic resources, because it is dictionary-independent.

Being statistical, our lemmatizer can only generate high quality lemmas through statistical features of its rules, and language specific parameters. Therefore the lemmatizer can generate noisy results, i.e. nonsense lemmas not belonging to the language. By parameter setting, StaLe can be made to emphasize either *lemmatization precision* or *lemmatization recall*.

We believe that lemmatization recall has priority because a human user can often recognize correct looking lemmas among nonsense. Also in automatic IR, when the database index and the queries are statistically lemmatized, we hypothesize that nonsense lemmas do not significantly deprave effectiveness because such lemmas in the database index do not frequently match nonsense query lemmas unless they originate from the same word form.

We show the performance of StaLe both in plain lemmatization tasks and as a component in an IR system using several datasets. For the lemmatization tests, we use the CLEF 2003 and PAROLE [14] collections to create the test word lists and a dictionary-based commercial lemmatizer TWOL [7] as the gold standard. It is as well-known and effective lemmatizer with large internal dictionaries for all the languages used in testing. We also experiment whether the gold standard can be improved when StaLe is used as an additional resource.

For the IR tests, we use CLEF 2003 full-text collections and the retrieval tool-kit Lemur, version 4.7 [9]. We couple with Lemur both StaLe and two state-of-the-art baselines, a lemmatizer and a stemmer. In the tests, we employ Finnish, Swedish, German and English, which represent morphological variation from highly complex to very simple. The main contribution of the present paper is to show the light-weight lemmatizer, StaLe, as an effective lemmatizer for low resource languages. This is why we test StaLe against standard techniques in high resource languages: only these languages allow using strong baselines. As the proposed method StaLe is shown

effective in handling this set of languages, we can trust that it is effective in handling other languages, e.g. ones with scarce resources.

The paper is structured as follows: in Chapter 2 we describe the statistical lemmatization method and discuss its working principles. We then present in Chapter 3 two kinds of test situations to measure the effectiveness of StaLe, and also describe how StaLe was set up for these tests. The results of the experiments are presented in Chapter 4 and discussed in Chapter 5 where we also make concluding remarks.

## 2. The StaLe Lemmatization Method

Our goal was to create a flexible and light, purely statistical lemmatizer, StaLe, which could operate with OOV words as well as with common vocabulary, and also be easily adaptable to new languages and domains. The lemmatizer was primarily aimed for languages with little resources for IR. Research and development in IR and practical retrieval in those languages would benefit from such tools.

StaLe produces the *result lemmas* for given *input token* by applying suitable rules from a *rule set* on the input token to create *candidate* lemmas. The candidate lemmas are then sorted by their confidence factor values and then pruned according to *parameter values* in the *candidate check-up phase*. The lemmas that qualify the check-up phase are the *result lemmas*. The parameters may also allow the input token itself to be added as a result lemma, because it may, with some probability, already be an ambiguous lemma. With a pre-generated rule database and trained parameter values the actual process of StaLe is quite simple as shown on the left side in Fig. 1.

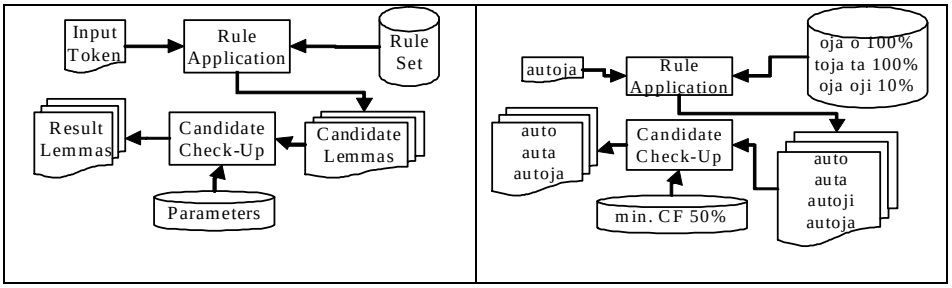


**Fig. 1.** The StaLe lemmatization process and an example process with input word “autoja”.

StaLe gives always at least one result word for input token. The input token itself is given as a result when no other candidate lemma can be created. An example how an input token is processed is given on the right side in Fig. 1. There an input token “autoja” (Finnish for “cars”, plural of partitive case) is the input. Three rules are found and applied to create the candidate lemma list including the input token itself. For example, the first rule suggest by 100% confidence that the ending ‘oja’ should be replaced by ‘o’. After the candidate check-up the lemma with lower confidence factor than defined in the parameters is dropped from the result list.

Our method is meant to serve primarily as a pure lemmatizer. *Compound words* are processed as *atomic* and not split, because corpus and dictionary independent compound splitting is highly ineffective. Airio [1] also showed that decompounding in monolingual IR is not vital.

Because StaLe is statistical and dictionary-independent, it may sometimes to create nonsense words in addition to correct lemma forms, and therefore lemmatization precision is poorer than when using dictionary-based methods. However, this can be tolerated in IR applications: finding the correct lemmas is primary and possible noise gets sorted out in other parts of IR.

### 2.1 Rules

The rules are a variation of *TRT-rules* by Pirkola and colleagues [16] with no restrictions on the length of the rule. A rule consists of *source* and *target strings*, the *position of the rule* and *statistic values* of the rule.

A rule string for English could, for example, be "fier fy e 8 100.0". The source string "fier" is the part of the unprocessed token that is replaced with the target string "fy" to form the candidate lemma. The rule also has a position value "e", which defines whether the rule applies in the beginning (b), in the end (e) or in the middle (m) of the given token. For some languages some of the positions are disabled, when those languages do not have word inflection in some positions.

Both strings in a rule share a *context character* that serves as a binding between the stem and affix of the word. The context character helps to prune the rule list and also separates rules dealing with consonant gradation from non-gradational rules.

Rules have two numerical values identical to the original TRT-rules. The first one is *rule frequency* representing the relative occurrence of the rule in the rule training list (in the example the rule frequency is the value "8"). The second value is *rule confidence factor* which measures how common the rule transformation is among all the transformations that match the same source string (value "100.0" in the example).

### 2.2 The Rule Training Lists and Parameter Tuning

A paired list of inflected tokens and their corresponding lemmas are needed to create the rules. The tokens and lemmas in the list should be representative for the language into which the method is applied and should be extracted from real texts. However, the list does not need to be an exhaustive representation of the language; quite the contrary: a small sample is enough.

Rules could be constructed intellectually or later hand-picked into the rule-set, but then they would lack first-hand information about the distribution and cross-relations between the rules and would lead into too severe deterioration of lemmatization precision with input data from real life materials. The statistical values are necessary for making distinctions between rules when processing words.

The size of the training list is relative to the morphological complexity of the target language [15]: the training list should be larger for languages with lots of morphological variation than for "simpler" languages. Usually, however, some

morphological features dominate while others are marginal and speculative, thus making it possible to gain good results without utilizing all possible rules [5]. Therefore it is possible to create a good set of rules using relatively small training list.

A key aspect for the applicability of StaLe is the one-time set-up effort it needs in the form of the training list. Lindén [10] estimates that two weeks of manual work can yield a list consisting about 30 000 token-lemma pairs. Our training lists contained roughly 10.000 to 30.000 training pairs, so this effort varies from a few days to two weeks of routine work; much less than needed for coding a new stemmer or a dictionary-based lemmatizer.

Lemmatization parameters reduce noise and improve lemmatization precision by defining a minimum rule confidence factor value for applied rules and controlling the number of created lemma candidates with an upper limit. Parameters are also needed to cope with situations where the input token already is a lemma and therefore should not be lemmatized, or the input token is not in the domain of the rules (i.e. if the rules are only for nouns and the input token is a verb). The parameters are strongly related to the rule sets and therefore should be trained using the same rules that are used in actual lemmatization. If a rule list is modified then it is appropriate to train a new set of parameters. The training is a matter of minor experiments.

## 3. Experiments

To assess the effectiveness of StaLe we did three kinds of performance tests. First we wanted to see how good the method is in plain lemmatization. This was done by processing paired token-lemma -wordlists with a program, which used StaLe to turn the tokens to candidate lemmas and compared those to the given lemmas.

Secondly we wanted to analyze how StaLe performs in IR when the document database index and the search topics are processed with StaLe. The results of these tests were compared against results achieved with a dictionary based gold standard lemmatizer and a stemmer in four languages.

Thirdly, we experimented with the significance of verbs in information retrieval tests. We examined typical CLEF 2003 query titles and descriptions and found that only about 5% of proper query words (i.e. topic words with stop-words pruned) are verbs. Bendersky and Croft [3] showed that specifically for long queries noun phrases are more effective that verb phrases. Nouns also carry most of the semantics in a query and represent the actual “things” that are retrieved [2], [11]. Therefore we assumed that the lack of verbs processing has at most a neglible effect on IR performance and to support our estimation we ran parallel IR tests with queries with verbs intact and with queries which had the verbs intellectually identified by their grammatical role in sentences and removed.

We used four languages, Finnish, Swedish, German and English, in all tests. For each language we trained rule sets and parameters which were used in all tests in each language. For all languages the input token itself was also included in the result list before the check-up phase to compete for listing in the final result.

### 3.1 Rule Lists

One rule set for each language was generated from parts of CLEF 2003 collections. Only nouns, adjectives and pronouns were used: verbs were excluded because the inflection of verbs differs significantly from the other word classes and would therefore increase the number of rules and excess noise.

For each of the four languages we varied the training list sizes to approximate the smallest set of rules that still would produce good results. The selected rule training list and corresponding rule list sizes are shown in Table 1. The estimated practical maximum of 30 000 word pairs was enough to produce a good rule set for Finnish, and the other languages needed much less to reach good enough rule sets.

**Table 1.** Rule training list sizes and corresponding rule set sizes.

| language | training word pairs | number of rules | language | training word pairs | number of rules |
|---|---|---|---|---|---|
| Finnish | 28610 | 5371 | German | 16086 | 426 |
| Swedish | 14384 | 960 | English | 9447 | 242 |

The sizes of the trained rule lists reflect the differences in the number of morphosyntactic features in various languages and the actual frequency of different inflectional word forms in rule training lists. English has only two features in grammatical case and German four, which explains the small size of the rule list. [15]

Parameters for each language were trained by testing with training lists generated with TWOL lemmatizers. The input token confidence factor values were selected from [6] and [15]: they are not related to any individual rule list, but to the language that the parameters and rule lists are used in, and do not therefore need training.

### 3.2 Lemmatization Tests

With lemmatization experiments we show how well StaLe lemmatizes inflected words and how well it treats words already in lemma form or words belonging to word classes that are outside the lemmatization rules. We used language specific versions of TWOL as the gold standard method and also a naïve baseline. In addition to testing StaLe and TWOL individually, we tested a combination in the “mixed” test setting (see below) where we lemmatized tokens with TWOL and treated the OOV tokens with StaLe. This is expected to maximize lemmatization performance.

For each language, we created three separate test lists from the CLEF collection for Finnish, PAROLE collection for Swedish and CLEF collections for German and English. The three lists for each language contained token-lemma pairs where the token was obtained from the collection and the lemma was either processed with TWOL or intellectually formed. To equalize the test situation for both lemmatizers, each list included only nouns, pronouns and adjectives. The number of words and the percentage of inflected words in the test lists are presented in Table 2.

**Table 2.** The number of words and the percentage of inflected words in the test lists.

| | Finnish | Swedish | German | English |
|---|---|---|---|---|
| **twolled** | 13825 (79%) | 15942 (60%) | 7834 (49%) | 1229 (19%) |
| **mixed** | 4792 (76%) | 1219 (52%) | 3894 (21%) | 2263 (1%) |
| **OOV** | 373 (100%) | 144 (99%) | 588 (32%) | 1084 (79%) |

Firstly, we formed a large test list using TWOL so that we could do a straight comparison between StaLe and the gold standard. This test list was created by extracting words recognized by TWOL in each language from text collections and identifying the lemmas by TWOL.

Secondly we created a mixed test list by lemmatizing the words with TWOL and then intellectually finding lemmas for words that were OOVs for TWOL. This test list resembled more a real-life situation where foreign words, unrecognized proper names, spelling errors and ad hoc words are present. The percentages of OOV-words in the mixed test list for Finnish are 22 %, for Swedish 23 %, for German 10 % and for English 37 %.

The third list contained only OOV words from the same text collections. The words were the words left unlemmatized by TWOL in the process of forming the first test lists. The test lists were finalized by intellectually processing the OOV material so that only inflected words were picked, because we wanted to analyze how well StaLe lemmatizes inflected words that are out of dictionary-based method's vocabulary.

### 3.3 IR Tests

IR tests were conducted to compare StaLe to the state-of-the-art baselines, and to investigate our assumptions that lemmatization recall has priority over lemmatization precision in an IR situation and that verbs are not essential for good performance.

We used CLEF 2003 full-text collections for each of the four languages and the query topics were also from CLEF 2003: 45 topics for Finnish, 54 for Swedish and English, and 56 for German. The IR system used to create the search indexes and to perform the retrieval operations was Lemur-toolkit version 4.7 into which the stemmer and both lemmatization methods were incorporated.

For each language, we built four indexes for the document collections: one with StaLe and the other three with the inflected baseline (marked *Bline*), the stemmer and the gold standard, respectively. The inflected baseline index was built without any morphological processing using the text tokens as they appeared. The stemmed index was created using the Snowball stemmers [18] for each language. Language specific versions of TWOL were used to return the gold standard lemmas. StaLe used the same rules and parameters as in the lemmatization tests. All the words in the documents were processed equally with the StaLe and thus also verbs, adverbs and particles were processed as nouns to evaluate a blind application of Stale.

Three types of queries were formed for each language. *Long queries* included the title and description fields of the query topics. *Long queries w/o verbs* were the same as the long queries, but with verbs intellectually removed prior to morphological processing. *Short queries* included only the title fields of the topics. Each query of

each of the three types was processed with the four methods and then matched only to indexes created with the same method.

### 3.4 Methods of Analysis

To assess the effectiveness of lemmatization we calculate lemmatization recall, lemmatization precision, $F_2$ values and mean values of those for each input token list. Lemmatization recall is the number of correct lemmas received with the input tokens inserted among the generated candidate lemmas. Lemmatization precision measures the proportion of false candidates in the result list.

The $F_2$-measure is a recall-biased derivate of the $F_n$-measure. F-measure combines recall and precision into one metric so that the results are easier to judge. The bias towards recall was made because lemmas not belonging to the language do not matter much in IR and are rather easy to distinguish in manual processes. For example, if an inflected word is translated into another language, the extra noise does probably not matter much because the translation dictionary eliminates the "non-words".

In the IR tests we measure effectiveness with *mean average precision* (MAP) and *precision at ten retrieved documents* (P@10). Statistical significance between the results of different methods was analyzed using *paired two-tailed t-test*. The standard significance level of 95% was selected as the limit for statistical significance.

## 4. Results

### 4.1 Lemmatization Tests

Table 3 gives the $F_2$ results for the lemmatization tests. The joint method of TWOL and StaLe is marked "T+S". The results between the baseline (Bline), on the one hand, and the two lemmatization methods, on the other, are all statistically significant and therefore the statistical significance is not indicated in Table 3. Note that for the gold standard (TWOL) the "twolled" word list necessarily yields F2 of 100%.

**Table 3.** The $F_2$-results of the lemmatization tests.

| language | Finnish | | | Swedish | | | German | | | English | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| test list | *twolled* | *mixed* | *OOV* | *twolled* | *mixed* | *OOV* | *twolled* | *mixed* | *OOV* | *twolled* | *mixed* | *OOV* |
| **T+S** | - | 93.75 | - | - | 95.45 | - | - | 95.26 | - | - | 98.50 | - |
| **TWOL** | 100 | 86.44 | 1.71 | 100 | 91.84 | 1.08 | 100 | 84.03 | 0.00 | 100 | 99.23 | 12.33 |
| **StaLe** | 67.74 | 68.73 | 58.19 | 76.48 | 78.18 | 67.01 | 90.72 | 88.62 | 74.55 | 91.09 | 95.82 | 84.45 |
| **Bline** | 24.51 | 23.85 | 0.25 | 42.10 | 48.54 | 0.39 | 70.46 | 60.95 | 0.00 | 81.10 | 98.88 | 12.33 |

In the TWOL-lemmatized test setting StaLe easily exceeds the baseline performance and reaches relatively close to the gold standard. In both Finnish and Swedish the test words were ambiguous and usually several suitable rules for each test word were found. The $F_2$ for Finnish is 67.74% and for Swedish 76.48%.

However, the lemmatization recalls are 88.6% and 96.4% respectively. With German and English the test list had a larger proportion of uninflected and unambiguous tokens which helped StaLe to reach $F_2$ scores around 91%.

With mixed test list TWOL's scores degraded at most 15.97 percent units (German) and only 0.77 percent units with English. However, TWOL returned the test word itself if no lemma could be found. When the baseline gave 98.88% for English, it is clear that the test list had only few inflected OOVs . Because StaLe created noisy results, it had a score below the baseline. Overall, StaLe's scores were stable across the test lists and for three languages StaLe was 20 to 74 % units above the baseline.

In Finnish the combination method improved the results by 7.31 percent units over plain TWOL. An improvement of 3.61 percent units was gained in for Swedish and 11.23 percent units in German. In English the results deteriorated 0.73 percent units. The inflected OOV word list naturally was nearly impossible for TWOL and baseline. However, StaLe was able to maintain its high level of performance.

### 4.2 IR Tests

The results for the IR tests are shown in Table 4, where the methods are sorted by their mean average precision (MAP) values. An asterisk indicates statistical significance in comparison with StaLe ($p < 0.05$). If the statistically significant value is smaller than the corresponding value of StaLe, then StaLe performed better in that test case, but if the statistically significant value is larger, then StaLe was inferior.

The results show that morphological processing has a strong effect on IR in Finnish improving MAP about 20 percent units from the baseline. StaLe and TWOL received quite similar results with no significant statistical difference: the scores of StaLe were from 92 % to 105 % of the TWOL scores. When verbs were included in long queries TWOL attained better MAP value, but when verbs were removed then StaLe was the winner. With short queries there was no significant difference between StaLe and TWOL. Stemming was nearly as effective as lemmatization in Finnish and the difference against StaLe was insignificant except in long queries without verbs.

In Swedish StaLe was at its weakest in the long queries with verbs, where the difference to TWOL was statistically significant. However, when the verbs were removed StaLe bypassed stemming and nearly caught TWOL narrowing the difference to insignificant. In short queries the differences were also insignificant.

**Long queries**

| *Finnish (N=45)* | | | *Swedish (N=54)* | | | *German (N=56)* | | | *English (N=54)* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** |
| TWOL | 52.76 | 35.11 | TWOL | 42.72* | 33.70 | TWOL | 45.04* | 48.57 | StaLe | 50.78 | 36.11 |
| StaLe | 48.69 | 33.11 | Stem | 40.48 | 32.22 | Stem | 42.70 | 48.39 | Stem | 48.06 | 35.74 |
| Stem | 46.33 | 30.89 | StaLe | 39.08 | 31.30 | StaLe | 41.40 | 46.61 | TWOL | 46.85* | 34.44 |
| Bline | 31.49* | 24.22* | Bline | 35.01* | 28.70 | Bline | 38.16 | 44.64 | Bline | 44.67* | 34.63 |

**Long queries w/o verbs**

| *Finnish (N=45)* | | | *Swedish (N=54)* | | | *German (N=56)* | | | *English (N=54)* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** |
| StaLe | 53.84 | 33.11 | TWOL | 43.07 | 33.33 | TWOL | 44.71* | 49.82 | StaLe | 48.38 | 36.11 |
| TWOL | 51.42 | 34.00 | StaLe | 42.40 | 33.52 | Stem | 42.49 | 48.75 | Stem | 47.25 | 35.19 |
| Stem | 47.89* | 30.44* | Stem | 41.11 | 31.85 | StaLe | 41.67 | 47.68 | TWOL | 45.58 | 35.00 |
| Bline | 38.95* | 26.22* | Bline | 37.11* | 29.26* | Bline | 37.86* | 45.00 | Bline | 44.18* | 33.70 |

**Short queries**

| *Finnish (N=45)* | | | *Swedish (N=54)* | | | *German (N=56)* | | | *English (N=54)* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** | **method** | **MAP** | **P@10** |
| TWOL | 45.43 | 29.78 | TWOL | 38.16 | 30.37 | TWOL | 35.56* | 44.64* | StaLe | 43.47 | 32.22 |
| StaLe | 45.42 | 31.11 | StaLe | 37.05 | 29.07 | Stem | 32.69 | 40.36 | TWOL | 42.52* | 30.74 |
| Stem | 39.37 | 27.27 | Stem | 36.12 | 28.52 | StaLe | 31.13 | 40.00 | Stem | 42.44 | 30.56 |
| Bline | 30.53* | 23.47* | Bline | 29.32* | 25.19* | Bline | 28.11* | 36.96* | Bline | 42.08 | 29.63 |

**Table 4.** The IR test results.

For German the morphological processing had similar effect as in Swedish. This time, however, StaLe could only reach roughly from 88 % to 93 % of the scores of TWOL. The difference between StaLe and stemming was statistically insignificant. Unlike in Finnish and Swedish, in German the inflected baseline was unaffected when verbs were removed.

The difference between the baseline and the gold standard was the smallest in English. Stemming was marginally better than TWOL, but StaLe was clearly the best method in short and long queries with verbs. Overall, StaLe's scores were from 102 % to 108 % of the gold standard scores. Unlike in other languages, StaLe's MAP results decreased when verbs were removed.

The exclusion of verbs did not seem to have a significant negative effect. On the contrary, the baseline MAPs improved in Finnish (by 7.46 percent units) and in Swedish (by 2.1 percent units) while the scores of morphological processing also improved with one exception: TWOL fared worse in Finnish. For German and English the scores diminished slightly when the verbs were excluded. Aside from the Finnish baseline, the differences between queries with and without verbs were statistically insignificant.

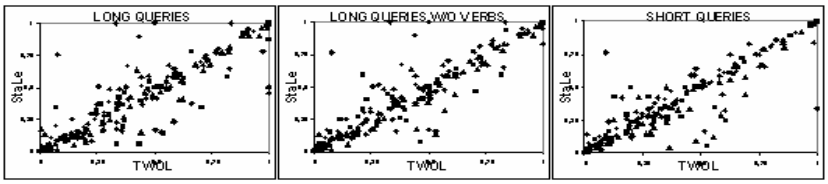


**Fig. 2.** Topic-by-topic differences between TWOL and StaLe.

Figure 2 illustrates the topic-by-topic difference between StaLe and TWOL. These two were chosen as they are the novel method and the gold standard, stemming winning StaLe only in 4/12 MAP cells, and TWOL in 2/12 MAP cells, of Table 4. In Fig. 2, each plot compares the MAP by StaLe to the MAP of TWOL across all four languages and all 209 topics. While there are deviations for the benefit of either lemmatizer, the data points clearly concentrate around the diagonal. This suggests equally robust performance for both lemmatizers and motivates the smallish differences between them in Table 4.

## 5. Discussion and Conclusions

We have described and tested a dictionary and corpus independent lemmatization method, StaLe, in lemmatization and information retrieval settings for four languages for which advanced morphological and lexical resources are available. The main findings of the tests indicate that StaLe is competitive with state-of-the-art methods used for comparison. This offers strong evidence for StaLe as an effective tool for low resource languages as well. Among the over 6000 living languages spoken globally,

the majority have poor language technology resources and are spoken in low resource communities. This makes StaLe attractive. In more detail the findings are as follows.

The strength of StaLe in the plain lemmatization tests was its robustness when more OOV words were introduced in the test word set. With the mixed test list StaLe still has $F_2$ below the gold standard lemmatizer because the mixed test list contained mostly common vocabulary, which is well-coded in the large dictionaries of the gold standard lemmatizer, TWOL. However, building a comprehensive dictionary for a new language is a great effort.

We also tested whether StaLe can improve the mixed test list results of TWOL. In this setting, TWOL processed all words it could and StaLe processed the leftover OOV words. For languages other than English this procedure proved effective improving results from 3.61 to 11.23 percent units. In English the score deteriorated below the baseline because inflected OOV words were very scarce in the test list and in those cases the StaLe gave a noisy result set. Adding StaLe to handle OOV words can clearly benefit the gold standard lemmatizer even in high resource languages.

StaLe had effectiveness equal to the dictionary-based gold standard method in IR tests. Morphologically more difficult languages required more rules in the StaLe rulebase. This increased the number of candidate lemmas and therefore lowered the precision scores somewhat. In English StaLe performed better than the competing methods because, firstly, with 242 rules StaLe was able to generalize the rare inflections and the verbs, and secondly, StaLe was able to process OOV words which usually are loan words and proper names.

In the IR tests the performance of the system using StaLe versus the system using the gold standard were within -4.5 to +4 percent units from each other, and also clearly above the baseline. Because StaLe can compete with state-of-the-art technologies in high resource languages, it is attractive to implement it for languages, which do not have such resources available.

Despite of the noise caused by the dictionary-independent approach, StaLe performed up to the dictionary-based gold standard. In plain lemmatization, noise lowered the $F_2$ score, but the IR tests showed that in practical applications noisy lemmas only have a negligible effect.

Our estimation on the negligible effect of the lack of verb processing proved correct: firstly, the removal of verbs from the IR queries did not have significant effects on performance, and performance was not significantly affected when verbs were included and processed as if being nouns. Note, that all document text tokens, including verbs, were indexed by StaLe as if being nouns. From this we conclude that verbs in general have no significant effect on IR and therefore they can be ignored when creating lemmatizers for document retrieval IR systems.

StaLe is easy to implement to new languages and it is robust enough to be effective with relatively little effort. A good rule-set requires at most only about 30 000 token-lemma pairs for morphologically complex language, which takes only a couple of weeks to construct, and less for easier languages. In addition to this, a similar but much smaller word list is also required for parameter training, but basically nothing else needs to be done to get StaLe working in a new language.

While StaLe is a pure lemmatizer it is possible to adjust it to give a probabilistic analysis of the input words because the rules already have values of a probabilistic

distribution over grammatical features (number, case, etc.). This could also lead to probabilistic surface syntax analysis. An additional dictionary structure could also be implemented to prune the nonsense lemmas. These belong to further work.

With $F_2$ score level near 70% for a morphologically complex language and around 90% for simpler ones in lemmatization tasks while also performing very well in retrieval tasks, StaLe is a light, robust and effective method for lemmatization.

## References

1. Airio, E.: Word normalization and decompounding in mono- and bilingual IR. Information Retrieval 9(3), pp. 249--271 (2006)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press/Addison-Wesley (1999)
3. Bendersky, M., Croft, W. B.: Analysis of Long Queries in a Large Scale Search Log. In: Proceedings of the 2009 workshop on Web Search Click Data, Barcelona, Spain, pp. 8--14 (2009)
4. Frakes, W. B., Baeza-Yates, R.: Information Retrieval: Data Structures & Algorithms. Prentice Hall, Englewood Cliffs, NJ, USA (1992)
5. Kettunen, K., Airio, E.: Is a morphologically complex language really that complex in full-text retrieval? In: Salakoski, T., et al. (Eds.), Advances in Natural Language Processing, LNAI 4139, pp. 411--422. Springer-Verlag, Berlin, Heiderberg (2006)
6. Kettunen, K.: Reductive and Generative Approaches to Morphological Variation of Keywords in Monolingual Information Retrieval. Acta Universitatis Tamperensis 1261. University of Tampere, Tampere (2007)
7. Koskenniemi, K.: Two-level Morphology: A General Computational Model for Word-Form Recognition and Production. Ph.D. Thesis, University of Helsinki, Department of General Linguistics, Helsinki (1983)
8. Krovetz, R.: Viewing morphology as an inference process. In: Proceedings of the 16th ACM/SIGIR Conference on Research and Development in Information Retrieval, pp. 191--202, Pittsburgh, Pennsylvania, USA (1993)
9. The Lemur Tool-kit for Language Modelling and Information Retrieval. http://www.lemurproject.org/ (visited 30.3.2010)
10. Lindén, K.: A Probabilistic Model for Guessing Base Forms of New Words by Analogy. In: Proceedings of CICling-2008, 9th International Conference on Intelligent Text Processing and Computational Linguistics, pp. 106--116, Haifa, Israel (2008)
11. Losee, R. M.: Is 1 Noun Worth 2 Adjectives? Measuring Relative Feature Utility. In: Information Processing and Management, 42(5), pp. 1248--1259 (2006)
12. Lovins, J. B.: Development of a Stemming Algorithm. Mechanical Translation and Computation Linguistics, 11(1), pp. 23--31 (1968)
13. Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P., Datta, K.: YASS: Yet Another Suffix Stripper. ACM Transactions on Information Systems (TOIS), 25(4) (2007)
14. Parole, Språkbanken, most common PAROLE words. http://spraakbanken.gu.se/eng/ (visited 30.3.2010)
15. Pirkola, A.: Morphological Typology of Languages for IR. Journal of Documentation, 57(3), pp. 330--348 (2001)
16. Pirkola, A., Toivonen, J., Keskustalo, H., Visala, K., Järvelin, K.: Fuzzy translation of cross-lingual spelling variants. In: Proceedings of the 26th ACM SIGIR Conference, pp. 345--353, Toronto, Canada (2003)

17. Plisson, J., Lavrac, N., Mladenic, D.: A rule based approach to word lemmatization. In: Proceedings of the 7th International Multi-Conference Information Society IS 2004, pp. 83--86 (2004)
18. Snowball. http://snowball.tartarus.org/ (visited 30.3.2010)
19. Wicentowski, R.: Modelling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework. Ph.D. Thesis, Baltimore, Maryland, USA (2002)