# Adaptive String Distance Measures for Bilingual Dialect Lexicon Induction

**Yves Scherrer**
Language Technology Laboratory (LATL)
University of Geneva
1211 Geneva 4, Switzerland
yves.scherrer@lettres.unige.ch

## Abstract

This paper compares different measures of graphemic similarity applied to the task of bilingual lexicon induction between a Swiss German dialect and Standard German. The measures have been adapted to this particular language pair by training stochastic transducers with the Expectation-Maximisation algorithm or by using hand-made transduction rules. These adaptive metrics show up to 11% F-measure improvement over a static metric like Levenshtein distance.

## 1 Introduction

Building lexical resources is a very important step in the development of any natural language processing system. However, it is a time-consuming and repetitive task, which makes research on automatic induction of lexicons particularly appealing. In this paper, we will discuss different ways of finding lexical mappings for a translation lexicon between a Swiss German dialect and Standard German. The choice of this language pair has important consequences on the methodology. On the one hand, given the sociolinguistic conditions of dialect use (diglossia), it is difficult to find written data of high quality; parallel corpora are virtually non-existent. These data constraints place our work in the context of scarce-resource language processing. On the other hand, as the two languages are closely related, the lexical relations to be induced are less complex. We argue that this point alleviates the restrictions imposed by the scarcity of the resources. In particular, we claim that if two languages are close, even if one of them is scarcely documented, we can successfully use techniques that require training.

Finding lexical mappings amounts to finding word pairs that are maximally similar, with respect to a particular definition of similarity. Similarity measures can be based on any level of linguistic analysis: semantic similarity relies on context vectors (Rapp, 1999), while syntactic similarity is based on the alignment of parallel corpora (Brown et al., 1993). Our work is based on the assumption that phonetic (or rather graphemic, as we use written data) similarity measures are the most appropriate in the given language context because they require less sophisticated training data than semantic or syntactic similarity models. However, phonetic similarity measures can only be used for *cognate language pairs*, i.e. language pairs that can be traced back to a common historical origin and that possess highly similar linguistic (in particular, phonological and morphological) characteristics. Moreover, we can only expect phonetic similarity measures to induce *cognate word pairs*, i.e. word pairs whose forms and significations are similar, as a result of a historical relationship.

We will present different models of phonetic similarity that are adapted to the given language pair. In particular, attention has been paid to develop techniques requiring little manually annotated data.

## 2 Related Work

Our work is inspired by Mann and Yarowsky (2001). They induce translation lexicons between a resource-rich language (typically English) and a scarce resource language of another language family (for example, Portuguese) by using a resource-

*Proceedings of the ACL 2007 Student Research Workshop*, pages 55–60,
Prague, June 2007. ©2007 Association for Computational Linguistics

rich bridge language of the same family (for example, Spanish). While they rely on existing translation lexicons for the source-to-bridge step (English-Spanish), they use string distance models (called *cognate models*) for the bridge-to-target step (Spanish-Portuguese). Mann and Yarowsky (2001) distinguish between *static metrics*, which are sufficiently general to be applied to any language pair, and *adaptive metrics*, which are adapted to a specific language pair. The latter allow for much finer-grained results, but require more work for the adaptation. Mann and Yarowsky (2001) use variants of Levenshtein distance as a static metric, and a Hidden Markov Model (HMM) and a stochastic transducer trained with the Expectation-Maximisation (EM) algorithm as adaptive metrics. We will also use Levenshtein distance as well as the stochastic transducer, but not the HMM, which performed worst in Mann and Yarowsky's study.

The originality of their approach is that they apply models used for speech processing to cognate word pair induction. In particular, they refer to a previous study by Ristad and Yianilos (1998). Ristad and Yianilos showed how a stochastic transducer can be trained in a non-supervised manner using the EM algorithm and successfully applied their model to the problem of pronunciation recognition (sound-to-letter conversion). Jansche (2003) reviews their work in some detail, correcting thereby some errors in the presentation of the algorithms.

Heeringa et al. (2006) present several modifications of the Levenshtein distance that approximate linguistic intuitions better. These models are presented in the framework of dialectometry, i.e. they provide numerical measures for the classification of dialects. However, some of their models can be adapted to be used in a lexicon induction task. Kondrak and Sherif (2006) use phonetic similarity models for cognate word identification.

Other studies deal with lexicon induction for cognate language pairs and for scarce resource languages. Rapp (1999) extends an existing bilingual lexicon with the help of non-parallel corpora, assuming that corresponding words share co-occurrence patterns. His method has been used by Hwa et al. (2006) to induce a dictionary between Modern Standard Arabic and the Levantine Arabic dialect. Although this work involves two closely related language varieties, graphemic similarity measures are not used at all. Nevertheless, Schafer and Yarowsky (2002) have shown that these two techniques can be combined efficiently. They use Rapp's co-occurrence vectors in combination with Mann and Yarowsky's EM-trained transducer.

## 3 Two-Stage Models of Lexical Induction

Following the standard statistical machine translation architecture, we represent the lexicon induction task as a two-stage model. In the first stage, we use the source word to generate a fixed number of candidate translation strings, according to a transducer which represents a particular similarity measure. In the second stage, these candidate strings are filtered through a lexicon of the target language. Candidates that are not words of the target language are thus eliminated.

This article is, like previous work, mostly concerned with the comparison of different similarity measures. However, we extend previous work by introducing two original measures (3.3 and 3.4) and by embedding the measures into the proposed two-stage framework of lexicon induction.

### 3.1 Levenshtein Distance

One of the simplest string distance measures is the Levenshtein distance. According to it, the distance between two words is defined as the least-cost sequence of edit and identity operations. All edit operations (insertion of one character, substitution of one character by another, and deletion of one character) have a fixed cost of 1. The identity operation (keeping one character from the source word in the target word) has a fixed cost of 0. Levenshtein distance operates on single letters without taking into account contextual features. It can thus be implemented in a memoryless (one-state) transducer. This distance measure is static – it remains the same for all language pairs. We will use Levenshtein distance as a baseline for our experiments.

### 3.2 Stochastic Transducers Trained with EM

The algorithm presented by Ristad and Yianilos (1998) enables one to train a memoryless stochastic transducer with the Expectation-Maximisation (EM) algorithm. In a stochastic transducer, all transitions represent probabilities (rather than costs or weights).

The transduction probability of a given word pair is the sum of the probabilities of all paths that generate it. The goal of using the EM algorithm is to find the transition probabilities of a stochastic transducer which maximise the likelihood of generating the word pairs given in the training stage. This goal is achieved iteratively by using a training lexicon consisting of correct word pairs. The initial transducer contains uniform probabilities. It is used to transduce the word pairs of the training lexicon, thereby counting all transitions used in this process. Then, the transition probabilities of the transducer are reestimated according to the frequency of usage of the transitions counted before. This new transducer is then used in the next iteration.

This adaptive model is likely to perform better than the static Levenshtein model. For example, to transduce Swiss German dialects to Standard German, inserting *n* or *e* is much more likely than inserting *m* or *i*. Language-independent models cannot predict such specific facts, but stochastic transducers learn them easily. However, these improvements come at a cost: a training bilingual lexicon of sufficient size must be available. For scarce resource languages, such lexicons often need to be built manually.

### 3.3 Training without a Bilingual Corpus

In order to further reduce the data requirements, we developed another strategy that avoided using a training bilingual lexicon altogether and used other resources for the training step instead. The main idea is to use a simple list of dialect words, and the Standard German lexicon. In doing this, we assume that the structure of the lexicon informs us about which transitions are most frequent. For example, the dialect word *chue 'cow'* does not appear in the Standard German lexicon, but similar words like *Kuh 'cow', Schuh 'shoe', Schule 'school', Sache 'thing', Kühe 'cows'* do. Just by inspecting these most similar existing words, we can conclude that *c* may transform to *k* (*Kuh, Kühe*), that *s* is likely to be inserted (*Schuh, Schule, Sache*), and that *e* may transform to *h* (*Kuh, Schuh*). But we also conclude that none of the letters *c, h, u, e* is likely to transform to *ö* or *f*, just because such words do not exist in the target lexicon. While such statements are coincidental for one single word, they may be sufficiently reliable when induced over a large corpus.

In this model, we use an iterative training algorithm alternating two tasks. The first task is to build a list of hypothesized word pairs by using the dialect word list, the Standard German lexicon, and a transducer[1]: for each dialect word, candidate strings are generated, filtered by the lexicon, and the best candidate is selected. The second task is to train a stochastic transducer with EM, as explained above, on the previously constructed list of word pairs. In the next iteration, this new transducer is used in the first task to obtain a more accurate list of word pairs, which in turn allows us to build a new transducer in the second task. This process is iterated several times to gradually eliminate erroneous word pairs.

The most crucial step is the selection of the best candidate from the list returned by the lexicon filter. We could simply use the word which obtained the highest transduction probability. However, preliminary experiments have shown that the iterative algorithm tends to prefer deletion operations, so that it will converge to generating single-letter words only (which turn out to be present in our lexicon). To avoid this scenario, the length of the suggested candidate words must be taken into account. We therefore simply selected the longest candidate word.[2]

### 3.4 A Rule-based Model

This last model does not use learning algorithms. It consists of a simple set of transformation rules that are known to be important for the chosen language pair. Marti (1985, 45-64) presents a precise overview of the phonetic correspondences between the Bern dialect and Standard German. Contrary to the learning models, this model is implemented in a weighted transducer with more than one state. Therefore, it allows contextual rules too. For example, we can state that the Swiss German sequence *üech* should be translated to *euch*. Each rule is given a weight of 1, no matter how many characters it concerns. The rule set contains about 50 rules. These rules are then superposed with a Levenshtein transducer, i.e. with context-free edit and identity opera-

[1] In the initialization step, we use a Levenshtein transducer.

[2] In fact, we should select the word with the lowest absolute value of the length difference. The suggested simplification prevents us from being trapped in the single-letter problem and reflects the linguistic reality that Standard German words tend to be longer than dialect words.

tions for each letter. These additional transitions assure that every word *can* be transduced to its target, even if it does not use any of the language-specific rules. The identity transformations of the Levenshtein part weigh 2, and its edit operations weigh 3. With these values, the rules are always preferred to the Levenshtein edit operations. These weights are set somewhat arbitrarily, and further adjustments could slightly improve the results.

## 4 Experiments and Results

### 4.1 Data and Training

Written data is difficult to obtain for Swiss German dialects. Most available data is in colloquial style and does not reliably follow orthographic rules. In order to avoid tackling these additional difficulties, we chose a dialect literature book written in the Bern dialect. From this text, a word list was extracted; each word was manually translated to Standard German. Ambiguities were resolved by looking at the word context, and by preferring the alternatives perceived as most frequent.[3] No morphological analysis was performed, so that different inflected forms of the same lemma may occur in the word list. The only preprocessing step concerned the elimination of morpho-phonological variants (*sandhi* phenomena). The whole list contains 5124 entries. For the experiments, 393 entries were excluded because they were foreign language words, proper nouns or Standard German words.[4] From the remaining word pairs, about 92% were annotated as cognate pairs.[5] One half of the corpus was reserved for training the EM-based models, and the other half was used for testing.

The Standard German lexicon is a word list consisting of 202'000 word forms. While the lexicon provides more morphological, syntactic and semantic information, we do not use it in this work.

The test corpus contains 2366 word pairs. 407 pairs (17.2 %) consist of identical words (lower bound). 1801 pairs (76.1%) contain a Standard German word present in the lexicon, and 1687 pairs (71.3%) are cognate pairs, with the Standard German word present in the lexicon (upper bound). It may surprise that many Standard German words of the test corpus do not exist in the lexicon. This concerns mostly *ad-hoc* compound nouns, which cannot be expected to be found in a Standard German lexicon of a reasonable size. Additionally, some Bern dialect words are expressed by two words in Standard German, such as the sequence *ir 'in the (fem.)'* that corresponds to Standard German *in der*. For reasons of computational complexity, our model only looks for single words and will not find such correspondences.

The basic EM model (3.2) was trained in 50 iterations, using a training corpus of 200 word pairs. Interestingly, training on 2000 word pairs did not improve the results. The larger training corpus did not even lead the algorithm to converge faster.[6] The monolingual EM model (3.3) was trained in 10 iterations, each of which involved a basic EM training with 50 iterations on a training corpus of 2000 dialect words.

### 4.2 Results

As explained above, the first stage of the model takes the dialect words given in the test corpus and generates, for each dialect word, the 500 most similar strings according to the transducer used. This list is then filtered by the lexicon. Between 0 and 20 candidate words remain, depending on how effective the lexicon filter has been. Thus, each source word is associated to a candidate list, which is ordered with respect to the costs or probabilities attributed to the candidates by the transducer. Experiments with 1000 candidate strings yielded comparable results.

Table 1 shows some results for the four models. The table reports the number of times the expected Standard German words appeared anywhere in the corresponding candidate lists (*List*), and the number

[3] Further quality improvements could be obtained by including the results of a second annotator, and by allowing multiple translations.

[4] This last category was introduced because the dialect text contained some quotations in Standard German.

[5] This annotation was done by the author, a native speaker of both German varieties. Mann and Yarowsky (2001) consider a word pair as cognate if the Levenshtein distance between the two words is less than 3. Their heuristics is very conservative: it detects 84% of the manually annotated cognate pairs of our corpus.

[6] This is probably due to the fact that the percentage of identical words is quite high, which facilitates the training. Another reason could be that the orthographical conventions used in the dialect text are quite close to the Standard German ones, so that they conceal some phonetic differences.

| | | N | L | P | R | F |
|---|---|---|---|---|---|---|
| Levenshtein | List | 840 | 3.1 | 18.5 | 35.5 | 24.3 |
| | Top | 671 | 1.1 | 32.7 | 28.4 | 30.4 |
| EM bilingual | List | 1210 | 4.5 | 21.4 | 51.1 | 30.2 |
| | Top | 794 | 0.7 | 52.5 | 33.6 | **41.0** |
| EM monolingual | List | 1070 | 5.0 | 16.6 | 45.2 | 24.3 |
| | Top | 700 | 0.7 | 47.9 | 29.6 | 36.6 |
| Rules | List | 987 | 3.2 | 22.8 | 41.7 | 29.5 |
| | Top | 909 | 1.0 | 45.6 | 38.4 | **41.7** |

Table 1: Results. The table shows the absolute numbers of correct target words induced (*N*) and the average lengths of the candidate lists (*L*). The three rightmost columns represent percentage values of precision (*P*), recall (*R*), and F-measure (*F*).

of times they appeared at the best-ranked position of the candidate lists (*Top*). Precision and recall measures are computed as follows:[7]

$$precision = \frac{|correct\ target\ words|}{|unique\ candidate\ words|}$$

$$recall = \frac{|correct\ target\ words|}{|tested\ words|}$$

As Table 1 shows, the three adaptive models perform better than the static Levenshtein distance model. This finding is consistent with the results of Mann and Yarowsky (2001), although our experiments show more clear-cut differences. The stochastic transducer trained on the bilingual corpus obtained similar results to the rule-based system, while the transducer trained on a monolingual corpus performed only slightly better than the baseline. Nevertheless, its performance can be considered to be satisfactory if we take into account that virtually no information on the exact graphemic correspondences has been given. The structure of the lexicon and of the source word list suffice to make some generalisations about graphemic correspondences between two languages. However, it remains to be shown if this method can be extended to more distant language pairs.

In contrast to Levenshtein distance, the bilingual EM model improves the *List* statistics a lot, at the expense of longer candidate lists. However, when comparing the *Top* statistics, the difference between the models is less marked. The rule-based model generates rather short candidate lists, but it still outperforms all other models with respect to the words proposed in first position. The rule-based model obtains high F-measure values, which means that its precision and recall values are better balanced than in the other models.

### 4.3 Discussion

All models require only a small amount of training or development data. Such data should be available for most language pairs that relate a scarce resource language to a resource-rich language. However, the performances of the rule-based model and the bilingual EM model show that building a training corpus with manually translated word pairs, or alternatively implementing a small rule set, may be worthwhile.

The overall performances of the presented systems may seem poor. Looking at the recall values of the *Top* statistics, our models only induce about one third of the test corpus, or only about half of the test words that *can* be induced by phonetic similarity models – we cannot expect our models to induce non-cognate words or words that are not in the lexicon (see the upper bound values in 4.1). Using the same models, Mann and Yarowsky (2001) induced over 90% of the Spanish-Portuguese cognate vocabulary. One reason for their excellent results lies in their testing procedure. They use a small test corpus of 100 word pairs. For each given word, they compute the transduction costs to each of the 100 possible target words, and select the best-ranked candidate as hypothesized solution. The list of possible target words can thus be explored exhaustively. We tested our models with Mann and Yarowsky's testing procedure and obtained very competitive results (see Table 2). Interestingly, the monolingual EM model performed much worse in this evaluation, a result which could not be expected in light of the results in Table 1.

While Mann and Yarowsky's procedure is very useful to evaluate the performance of different similarity measures and the impact of different language pairs, we believe that it is not representative for the task of lexicon induction. Typically, the list of possible target words (the target lexicon) does not contain 100 words only, but is much larger (202'000 words in our case). This difference has several implications. First, the lexicon is more likely to present very

[7] The words that occur in several candidate lists (i.e., for different source words) are counted only once, hence the term *unique candidate words*.

| | Mann and Yarowsky | | Our work | |
|---|---|---|---|---|
| | cognate | full | cognate | full |
| Levenshtein | 92.3 | 67.9 | 90.5 | 85.2 |
| EM bilingual | 92.3 | 67.1 | 92.2 | 86.5 |
| EM monolingual | | | 81.9 | 76.7 |
| Rules | | | 94.1 | 88.7 |

Table 2: Comparison between Mann and Yarowsky's results on Spanish-Portuguese (68% of the full vocabulary are cognate pairs), and our results on Swiss German-Standard German (83% cognate pairs). The tests were performed on 10 corpora of 100 word pairs each. The numbers represent the percentage of correctly induced word pairs.

similar words (for example, different inflected forms of the same lexeme), increasing the probability of "near misses". Second, our lexicon is too large to be searched exhaustively. Therefore, we introduced our two-stage approach, whose first stage is completely independent of the lexicon. The drawback of this approach is that for many dialect words, it yields no result at all, because the 500 generated candidates were all non-words. The recall rates could be increased by generating more candidates, but this would lead to longer execution times and lower precision rates.

## 5 Conclusion and Perspectives

The experiments conducted with various adaptive metrics of graphemic similarity show that in the case of closely related language pairs, lexical induction performances can be increased compared to a static measure like Levenshtein distance. They also show that requirements for training data can be kept rather small. However, these models also show their limits. They only use single word information for training and testing, which means that the rich contextual information encoded in texts, as well as the morphologic and syntactic information available in the target lexicon, cannot be exploited. Future research will focus on integrating contextual information about the syntactic and semantic properties of the words into our models, still keeping in mind the data restrictions for dialects and other scarce resource languages. Such additional information could be implemented by adding a third step to our two-stage model.

## Acknowledgements

We thank Paola Merlo for her precious and useful comments on this work. We also thank Eric Wehrli for allowing us to use the LATL Standard German lexicon.

## References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

Wilbert Heeringa, Peter Kleiweg, Charlotte Gooskens, and John Nerbonne. 2006. Evaluation of string distance algorithms for dialectology. In *Proceedings of the ACL Workshop on Linguistic Distances*, pages 51–62, Sydney, Australia.

Rebecca Hwa, Carol Nichols, and Khalil Sima'an. 2006. Corpus variations for translation lexicon induction. In *Proceedings of AMTA'06*, pages 74–81, Cambridge, MA, USA.

Martin Jansche. 2003. *Inference of String Mappings for Language Technology*. Ph.D. thesis, Ohio State University.

Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the ACL Workshop on Linguistic Distances*, pages 43–50, Sydney, Australia.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL'01*, Pittsburgh, PA, USA.

Werner Marti. 1985. *Berndeutsch-Grammatik*. Francke Verlag, Bern, Switzerland.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of ACL'99*, pages 519–526, Maryland, USA.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of CoNLL'02*, pages 146–152, Taipei, Taiwan.