Multilingual Modeling of Cross-Lingual Spelling Variants

Krister Lindén
*Helsinki University, Department of General Linguistics, P.O.Box 9 (Siltavuorenpenger 20 A), FIN-00014 University of Helsinki, Finland, (*krister.linden@helsinki.fi*)*

Abstract.

Technical term translations are important for cross-lingual information retrieval. In many languages, new technical terms have a common origin rendered with different spelling of the underlying sounds, also known as cross-lingual spelling variants (CLSV).

To find the best CLSV in a text database index, we contribute a formulation of the problem in a probabilistic framework, and implement this with an instance of the general edit distance using weighted finite-state transducers. Some training data is required when estimating the costs for the general edit distance. We demonstrate that after some basic training our new multilingual model is robust and requires little or no adaptation for covering additional languages, as the model takes advantage of language independent transliteration patterns.

We train the model with medical terms in seven languages and test it with terms from varied domains in six languages. Two test languages are not in the training data. Against a large text database index, we achieve 64–78 % precision at the point of 100 % recall. This is a relative improvement of 22 % on the simple edit distance.

Keywords: Term translations, Cross-lingual information retrieval, Systematic spelling variants, General edit distance

1. Introduction

Finding term translations as cross-lingual spelling variants on the fly is an important problem for cross-lingual information retrieval (CLIR). CLIR is typically approached by automatically translating a query into the target language. For an overview of the approaches to cross-lingual information retrieval, see (Oard and Diekema, 1998). When automatically translating the query, specialized terminology is often missing from the translation dictionary. The analysis of query properties in (Pirkola and Järvelin, 2001) shows that proper names and technical terms often are prime keys in queries, and if not properly translated or transliterated, query performance may deteriorate significantly. As proper names often need no translation in languages using the same writing system, a trivial solution is to include the untranslatable keys as such into the target language query. However, technical terms often have common roots, which allows for a more advanced solution using approximate string matching[1] to find the target words, most similar to the source keys, in the index of the target language text database (Pirkola et al., 2001).

In European languages, the loan words are often borrowed with minor, but language specific, modifications of the spelling of a common Greek or Latin root. This allows for initial testing of some straight forward approximate

string matching methods. A comparison of methods applied to cross-lingual spelling variants in CLIR for a number of European languages is provided in (Keskustalo et al., 2003). They compare exact match, simple edit distance, longest common subsequence, digrams, trigrams, tetragrams, as well as skipgrams, i.e., digrams with gaps. Skipgrams perform best in their comparison with a relative improvement of 7.5 % on the average on the simple edit distance baseline. They also show that among the baselines, the simple edit distance baseline is in general the hardest baseline to beat. They use no explicit n-gram transformation information. Such transformations are used in (Pirkola et al., 2003), where they are based on digrams and trigrams. Trigrams are better than digrams, but they make no comparison to the edit distance baseline. However, in both of the previous studies on European languages, the distance measures based on n-grams use a bag of n-grams ignoring their sequential order.

Between languages with different writing systems, foreign words are often borrowed based on phonetic rather than orthographic transliterations suggesting a phoneme-based rather than a grapheme-based approach. In (Knight and Graehl, 1998), a phoneme-based generative model is introduced which transliterates words from Japanese to English using weighted finite-state transducers. In (Qu et al., 2003), this approach is successfully evaluated in an information retrieval context. This model uses context-free transliterations, which produces heavily overgenerating systems. Context-sensitivity requires more training data, but training data is less readily available for phoneme-based approaches, which lately have been rivaled by grapheme-based approaches, e.g., to Arabic (Al-Onaizan and Knight, 2002), Japanese (Ohtake et al., 2004; Bilac and Tanaka, 2004), and Chinese (Zhang et al., 2004). Until now, such models have included only one language pair.

Our first contribution is to present a formulation of finding *cross-lingual spelling variants in a probabilistic framework*. The second contribution is to reformulate this as an *instance of the general edit distance* and to show how this is *efficiently implemented with weighted finite-state transducers* using context-sensitive transliterations. The costs for the general edit distance are learned from a training sample of term pairs. The third contribution is to demonstrate that a distance measure, which explicitly accounts for the *sequential order of the n-grams*, significantly *outperforms* models based on *unordered bags of n-grams*. The final contribution of this article is to demonstrate that our model needs *little or no adaptation* for covering new language pairs and that the *model is robust*, i.e., adding a new language does not adversely affect the performance of the model for the already trained languages.

In our first experiment, we train and test a multilingual model with terms from the medical domain. Against an index of a large English newspaper database, we achieve 80–91 % precision at the point of 100 % recall for a set

of medical terms in Danish, Dutch, French, German, Italian, Portuguese and Spanish. On the average, this is a relative improvement of 26 % on the simple edit distance baseline. In our second experiment, we use the medical terms as training data and test with a set of terms from varied domains. We achieve 64–78 % precision at the point of 100 % recall in French, German, Italian, Spanish, Swedish and Finnish. On the average, this is a relative improvement of 22 % on the simple edit distance baseline. For Swedish, there is no training data, and for Finnish, we need only a small amount of training data for adapting the multilingual model, which demonstrates that the model has captured essential language independent transliteration patterns.

The rest of this article is organized as follows. Sect. 2 *Methodology* introduces the probabilistic framework and outlines its implementation with weighted finite-state transducers. Sect. 3 *Data Sets* presents the training, test and adaptation data collections as well as the baselines. In Sect. 4 *Experiments*, we present the experiments and evaluate the results and their importance. In Sect. 5 *Discussion*, we discuss the linguistic motivations for the model and some related work.

2. Methodology

First we describe a method for finding the best cross-lingual spelling variants (CLSV) for a given search key with an unknown translation and present it in terms of a probabilistic framework. We then outline how the framework can be implemented with a cascade of weighted finite-state transducers.

2.1. Probabilistic Framework

Assume that we have a word in a foreign language. We call this the source word S. The word looks familiar and we want to know the possible meanings of this word in a language known to us, but we do not have a translation dictionary, only a word list of the words in the known language. We take the word and compare it to all the words in the word list L in order to determine which word is most similar to the unknown word. We call the most similar word the target word T. In the beginning we can only compare how many letters are similar or different, but having done so a few times, we start to learn the regularities involved where some letters are likely to be inserted, deleted or replaced with others. We also observe that the likelihood for insertion, deletion and replacement for each letter is different in different contexts.

A model for this procedure is to find, in the target word list L, the target word T which is most likely to convey the meaning of the source word S by accounting for all the sounds encoded by the source word letter sequence. To find the most likely target word for any given source word, we need to maximize the probability $P(T|S)$, i.e.,

$$\arg\max_{T \in L} P(T|S)\,. \tag{1}$$

In order to automate this procedure we take a sample of source words, whose target words we have already determined. We then align the source words and their target words using the minimum simple edit distance assuming that most of the corresponding letters of the alphabets of the two languages represent roughly the same sounds. For the simple edit distance, the cost of editing the original string with a replacement, insertion or deletion is one, whereas keeping the same letter has zero cost. In Table I, we see an example of two words aligned with the minimum simple edit distance.

Table I. Minimum edit distance alignment of the target word *capacity* in English and source words *capacidad* in Spanish at cost 3 and *Kapazität* in German at cost 4, when ϵ is the empty string

Src		Trgt		Src		Trgt	
c		c		K	↔	c	(rep)
a		a		a		a	
p		p		p		p	
a		a		a		a	
c		c		z	↔	c	(rep)
i		i		i		i	
d	↔	t	(rep)	t		t	
a	↔	y	(rep)	ä	↔	y	(rep)
d	↔	ϵ	(del)	t	↔	ϵ	(del)

From the set of source words aligned with their target words, we derive the frequency of each edit operation in context. We take into account a context of at most four letters, cf. Sect. 5.1 *Cross-lingual Spelling Changes*, in the source word S including the letter s_i aligned with the letter t_i in the target word T. Fixing the alignment of the target and source word letters requires that we also consider the possibility that a target word letter is aligned with no letter at all, i.e., the empty string, in the source word and vice versa. We need to consider at most $max(|T|, |S|)$ positions in the target word T, $|S| - |T|$ of which are empty strings if $|S| > |T|$. If we fix the context of the letter s_i, e.g., to be one letter to the left and two to the right, we need to pad each word with a placeholder at the beginning and two at the end of the word extending the alphabet with a padding letter. We use # as a padding letter.

We denote the context $s_{i-1}s_is_{i+1}s_{i+2}$ in the source word with S_{i4}, where s_i occurs after s_{i-1} and before s_{i+1} and s_{i+2} in the source word, and s_{i-1},

s_{i+1} and s_{i+2} can be any letters of the source language alphabet. The expression $t_i|S_{i4}$ can be seen as a transformation of s_i into t_i in the context S_{i4}. Here s_i and t_i can be any letter of the source and target language alphabet, respectively, as well as the empty string ϵ. In the Equation

$$P(T|S) = \prod_{i=1..max(|T|,|S|)} P(t_i|S_{i4})\,, \tag{2}$$

the probability of the transformation $t_i|S_{i4}$ is estimated with the count for the transformation divided by the count for the context S_{i4}, i.e., the probability $P(t_i|S_{i4}) \cong cnt(t_i|S_{i4})/cnt(S_{i4})$. This defines a probability distribution for the transformations of s_i into t_i in the source word context S_{i4}.

If a context occurs too seldom, e.g., less than M times, cf. Sect. 4.1 *Experiments on Training Data*, the reliability of the estimate for the probability distribution is low. We use an offline back-off model for smoothing the probability $P(t_i|S_{i4})$. We define the contexts $S_{i3} = s_{i-1}s_is_{i+1}$, $S_{i2} = s_{i-1}s_i$ and $S_{i1} = s_i$. The back-off model is defined as

$$P(t_i|S_{i4}) \cong \begin{cases} cnt(t_i|S_{i4})/cnt(S_{i4}) \text{ if } cnt(S_{i4}) \geq M, \\ cnt(t_i|S_{i3})/cnt(S_{i3}) \text{ if } cnt(S_{i4}) < M \wedge cnt(S_{i3}) \geq M, \\ cnt(t_i|S_{i2})/cnt(S_{i2}) \text{ if } cnt(S_{i3}) < M \wedge cnt(S_{i2}) \geq M, \\ cnt(t_i|S_{i1})/cnt(S_{i1}) \text{ if } cnt(S_{i2}) < M. \end{cases} \tag{3}$$

We use Laplace's law for successions for discounting unseen transformations with the additional assumption that a letter is most likely to remain untransformed in any given context. Let $|A|$ be the size of the target language alphabet. For each j, when $j = 1 \ldots 4$, the transformation count $cnt(t_i|S_{ij})$ is increased by $1/2$ if $s_i = t_i$ and by $1/2 * (|A| - 1)^{-1}$ if $s_i \neq t_i$, and the context count $cnt(S_{ij})$ is increased by 1. For unseen contexts, this gives 50 %[2] of the probability mass to keeping a letter untransformed, and the rest is evenly distributed among the transformations to other letters or the empty string, thus roughly modeling the simple edit distance.

The target word T with the highest probability in the target word list L is

$$\arg\max_{T \in L} P(T|S) = \arg\max_{T \in L} \prod_{i=1..max(|T|,|S|)} P(t_i|S_{i4})\,. \tag{4}$$

2.2. WEIGHTED FINITE-STATE TRANSDUCERS

Finding the CLSVs or target words with the highest probabilities can be efficiently implemented with a cascade of finite-state transducers[3] composed into a *Translation* transducer.

First the source word S is expanded into an automaton of tetragrams using a transducer called *Tetrify*:

$$b \rightarrow (a\epsilon bc)^* abcd / a \_ c\ d, \tag{5}$$

i.e., b is replaced with the regular expression of tetragrams $(a\epsilon bc)^* abcd$, when a precedes b and b is followed by c and d in the source word, where a, b, c and d represent any letter of the source language alphabet. The tetragrams are letters of the alphabet in the new automaton. The tetragrams with the ϵ symbol in (5) are used for introducing the positions of empty strings in the source word that may be aligned with letters in the target word.

If a tetragram is too infrequent, i.e., it occurs less than M times, the back-off model from (3) is implemented by replacing the tetragram with a trigram, digram or unigram representing a more general and sufficiently frequent context. In a transducer called *Backoff*, each n-gram symbolizes a source word letter context:

$$axcd \rightarrow \begin{cases} axcd & \text{if } cnt(axcd) \geq M, \\ axc & \text{if } cnt(axcd) < M \wedge\ cnt(axc) \geq M, \\ ax & \text{if } cnt(axc) < M \wedge\ cnt(ax) \geq M, \\ x & \text{if } cnt(ax) < M, \end{cases} \tag{6}$$

where x may be any letter of the source language alphabet as well as the empty string ϵ.

A *Weight* transducer represents the probability distribution of the target language letters for each source letter context. The transducer is implemented in the tropical semi-ring giving each transduction a log-probability weight:

$$\begin{array}{ll} axcd \rightarrow y & \text{with } -log(P(y|axcd)), \\ axc \rightarrow y & \text{with } -log(P(y|axc)), \\ ax \rightarrow y & \text{with } -log(P(y|ax)), \\ x \rightarrow y & \text{with } -log(P(y|x)), \end{array} \tag{7}$$

where y is any letter of the target language alphabet as well as the empty string ϵ. The target word list is compiled into an identity transducer called *Targetindex*. All the possible target words T which correspond to a source word S are found by composing the cascade of transducers into the *Translation* transducer:

$$\textit{Translation} = \textit{S o Tetrify o Backoff o Weight o Targetindex}\,. \tag{8}$$

To extract the target words T, we make a projection of the *Translation* transducer on the target word surface. The N-best target words of the projection are listed, i.e., the N target words with the smallest total log-probability weights.[4]

The process is outlined in Table II for the transliteration of the German word *Kapazität* into the English word *capacity*. The table shows all the alternatives of the fully trained model. The transductions are unambiguous until

the *Weight* transductions. The context-sensitivity leaves fairly few options to be considered and only for *azi* is the correct alternative the second best. For brevity, we did not include in the table that *Tetrify* introduces a placeholder context allowing insertions between every letter of the source word: E.g., between the source word letters *K* and *a*, there would be the placeholder context *kϵap* corresponding in this case to an empty string in the target word.

3. Data Sets

First we describe the training, test and adaptation data collections. We then motivate our choice of baseline for the task of finding CLSVs for words with unknown translations and present a baseline for our method on the training and test data collections.

3.1. TRAINING DATA

The problem of finding terms for unknown words crops up in many different areas with quickly developing terminologies, e.g., medicine, economics, technology. As training data for our model we chose technical medical terminology in seven different languages as source words. We chose English as the target language. The terminology was extracted from the web pages of the EU project *Multilingual Glossary of Technical and Popular Medical Terms in Nine European Languages* by (Stichele, 1995). Only eight languages were available on the web server: Danish, Dutch, English, French, German, Italian, Portuguese and Spanish. We collected 1617 words which had at least one translation in all eight languages.

3.2. TEST DATA

To be able to compare our test results, we used the test data created by (Keskustalo et al., 2003) at the University of Tampere. The test data consists of three parts: the target words, the search keys, and the set of correct answers (relevance judgments). Their characteristics are recapitulated below.

The target words consist of a list containing all words of an English full-text database index of 84 000 documents (Los Angeles Times used in the CLEF 2000 experiments) (Peters, 2000). It contains around 189 000 unique word forms. The words are either in base form if recognized by the morphological analyzer ENGTWOL (Voutilainen et al., 1995) used in indexing, or in case of unrecognized word forms, the original words as such. All words are written in monocase.

The 271 search keys are translations into six languages of terms selected from the English database index. The terms did not occur in a standard translation dictionary. The terms can be grouped into domains. The number of

Table II. Transliteration of *Kapazität* in German to *capacity* in English with the *Translation* transducer. The empty string is symbolized by ϵ. A smaller weight means a higher probability

S	Tetrify	Backoff	Weight	Matching entries in Targetindex
#	ϵ#ka	ϵ#ka	#/0.00	#
K	#kap	#kap	c/0.22 k/1.61	c
a	kapa	kap	a/0.00	a
p	apaz	apa	p/0.00	p
a	pazi	pa	a/0.01 i/6.39 ϵ/6.39	a
z	azit	azi	t/0.14 c/2.44 ϵ/3.39 z/5.00 s/5.00	c
i	zitä	zitä	i/0.00	i
t	ität	ität	t/0.05 e/3.64 n/3.64	t
ä	tät#	tät#	y/0.08 s/3.66 e/3.66	y
t	ät##	ät##	ϵ/0.08 s/2.97 t/3.66	ϵ
#	t##ϵ	t##ϵ	#/0.00	#
#	##$\epsilon\epsilon$	##$\epsilon\epsilon$	#/0.00	#

terms in English in each domain is indicated in parenthesis: medicine or biology (90), geographical place names (31), economics (55), technology (36) and others (59). The terms were translated into Finnish, French, German, Italian, Spanish and Swedish. For some terms, there are more than one translation resulting in a slightly varying number of search keys for each test language.

For each translated term, its English equivalent was considered the correct answer, i.e., the relevant target word to be identified.

3.3. Adaptation Data

Based on the 1617 English training data terms, we created adaptation data for Finnish by consulting several online resources. The most significant resource was the online medical language database *Tohtori.fi – Lääkärikirja* (Nienstedt, 2003). We found Finnish equivalents for 1480 of the medical terms.

3.4. Baselines and Significance Tests

The ideal method for this task should always give the correct translation or transliteration as the first choice. In this case the precision would be 100 % at the level of 100 % recall. Occasionally, non-ideal methods will give x incorrect suggestions before the correct one. In this case the precision is $1/(1+x)$ at 100 % recall. If the correct answer is not among the candidates suggested by a method for a certain test word, the precision is 0 % at 100 % recall for this test word. The overall performance of a method is the average precision at 100 % recall. Using the precision of each test word, we can calculate the average precision and the standard deviation. We measure the significance of improvements in the performance with the z-test, which compares two sample means to suggest whether both samples come from the same population (Kanji, 1999).

In order to evaluate the performance of our method on the training data, we chose the edit distance as a baseline. This is motivated by the research done by (Keskustalo et al., 2003), which shows that the edit distance is often the most difficult baseline to beat for this type of task among a number of other baselines.

For the training data, the performance of the simple edit distance is shown in Table III. The table also shows the average edit distance and the percentage of exact matches. The standard deviation of the baseline in each language is approximately 1.0 %. The average baseline for all the languages is 67.7 $\pm$0.4 %. An exact match has simple edit distance zero. The percentage of exact matching terms for each of the languages with regard to the English terminology is fairly low, i.e., on the average 17.1 % of the 1671 terms.

Table III. The edit distance baseline, the average minimum edit distance and the percentage of exact matches for each language in the training data

Language	Edit Distance Baseline	Average Edit Distance	Exact Matches
Danish	71.0	1.86	28.1
Dutch	67.2	1.80	17.8
French	67.9	1.64	19.4
German	73.2	1.84	25.0
Italian	62.7	2.37	4.5
Portuguese	59.7	2.29	10.0
Spanish	72.5	1.75	15.1
Average	67.7	1.94	17.1

For the test data, the performance of the simple edit distance and the best skipgrams are shown in Table IV. The table also shows the percentage of exact matches. The numbers in the table are from (Keskustalo et al., 2003). We grouped the table according to languages present in the training data. The standard deviation of the baseline in each language is approximately 2.5 %. The average simple edit distance baseline for all the languages is 57.5 $\pm$1.2 %. The percentage of exact matching terms for each language is on the average 17.7 % of the 271 English terms.

The percentage of exact matches for French in the training data may seem relatively low compared to that in the test data. This is due to the differing principles for compiling the data collections. The EU project selected preferred or recommended term translations, whereas (Keskustalo et al., 2003) selected those term translations that are CLSVs. The percentage of exact matches for Finnish in the test data is very low due to the Finnish syllable structure, which almost always requires additional characters in words and terminology borrowed from other languages.

4. Experiments

We did initial experiments with the training data in order to tune the parameters and gain reference values for the performance level improvements. We then applied the trained model to the test data. Finally, we studied the performance of our model when adapting it to an additional language.

Table IV. The simple edit distance, the best skipgram result and the percentage of exact matches for each language in the test data

Language	Edit distance Baseline	Skipgram Precision	Exact Matches
French	72.2	75.5	40.8
German	60.8	65.7	17.6
Italian	53.2	57.2	12.8
Spanish	57.0	60.0	13.7
Finnish	45.9	49.9	1.8
Swedish	56.0	62.1	19.7
Average	57.5	61.7	17.7

4.1. Experiments on Training Data

In our experiments on the training data, we used part of the training data to train the general edit distance model and the rest to evaluate how well the model was able to pick the correct translation from the English database index. We used 10-fold cross-validation on the training data, i.e., we estimated the parameters on 90 % of the terminology and used 10 % for testing. Each time we used a different portion of the training data for training and testing purposes.

When training, we pooled the training data, i.e., we derived context and transformation counts from all the languages, paired with English as the target language, cf. Sect. 5.2 *Multi-lingual Modeling*. In an initial experiment, we determined that pooling the training data gives a statistically significantly better performance than training separately for each individual language pair. The initial experiments also showed that using a minimum context frequency $M = 4$ from a range of $1 \ldots 6$ in the back-off model yielded the best transliteration results on the training data.

The model is relatively fast. We achieved a performance average of approximately 0.58 seconds per source word for computing all possible transliterations in the target word index containing approximately 189 000 word forms. We used an Intel Pentium 4 with 1.8 GHz CPU and 1 GB of memory. The speed is crucially dependent on the context-sensitive transformations allowing few and correct transformations in each context. Insertions are relatively rare in most contexts, but the back-off model easily backs off to a broad range of context-free insertions. Allowing the model to explore

context-free insertions increases the time consumption 8.5-fold, but adds only 1.0 % to the average precision. We disallowed backing off to context-free insertions. Further speed-up could have been achieved by generally pruning low-probability transformations.

The overall result for all the languages was 85.4 ±0.3 % with our method. This is a relative improvement of 26.5 % on the simple edit distance baseline. The average precision as well as the standard deviation for each language is shown in Table V. The average precision is compared to that of the baseline and the absolute and relative performance improvement is calculated.

Table V. The average precision on the training data for all the languages using English as the target language as well as the relative and absolute improvement

Language	Baseline	Average Precision	Standard Deviation	Absolute Improvement	Relative Improvement
Danish	71.0	86.5	±0.8	15.5	21.8
Dutch	67.2	90.6	±0.7	23.4	34.8
French	67.9	86.9	±0.8	19.0	28.0
German	73.2	88.7	±0.7	15.6	21.3
Italian	62.7	80.2	±0.9	17.6	28.0
Portuguese	59.7	82.0	±0.9	22.3	37.3
Spanish	72.5	82.9	±0.9	10.4	14.3
Average	67.7	85.4	±0.3	17.7	26.5

With English as the target language, the languages from the Germanic language family, i.e., German, Dutch and Danish, perform in the range of 86–91 %, whereas the languages from the Romance language family, i.e., French, Italian, Portuguese and Spanish, perform in the range of 80–87 % on the training data. This is very highly significantly better than the baselines with $p<0.0001$.

4.2. Experiments on Test Data

For the experiments on the test data we used a model that was trained on all of the pooled training data. The average precision as well as the standard deviation for each test language is shown in Table VI. The average precision is compared to that of the baseline and the absolute and relative performance improvement is calculated.

The languages that were present in the training data improved significantly as expected, except French, which improved less than expected. There was no training data for Finnish and Swedish. Finnish from the Fenno-Ugric lan-

guage family performed below the baseline. Swedish, however, was in line with German from the same Germanic language family.

Table VI. The test results for each of the languages on the test data without training or adaptation data for Swedish and Finnish

Language	Baseline	Precision	Standard Deviation	Absolute Improvement	Relative Improvement
French	72.2	77.1	±2.3	4.9	6.8
German	60.8	73.2	±2.4	12.4	20.4
Italian	53.2	65.7	±2.6	12.5	23.6
Spanish	57.0	65.3	±2.6	8.3	14.6
Finnish	45.9	40.7	±2.8	-5.2	-11.3
Swedish	56.0	73.5	±2.5	17.5	31.2

4.3. Experiments with Adaptation Data

We wanted to study the robustness of our model when adapting it to a new language. As the results from the experiments on the test data suggested, no further adaptation was needed when adding a language from a language family present in the training data, so we studied a language from a new language family by adapting the model to Finnish from the Fenno-Ugric language family.

We gradually added adaptation data for Finnish and observed the performance of the model on the test data. We did 10 test runs with random permutations of the adaptation data. The result is shown in Fig. 1. When adding approximately 19 randomly selected Finnish-English term pairs to the training data, we reached the baseline. When adding approximately 37 randomly selected term pairs, we reached the skipgram performance. After approximately 150 term pairs no further statistically significant improvement could be observed with the test data.

The final overall result for all the languages was 70.2 ±1.0 % when the model was fully adapted to Finnish. This is very highly significantly better than the baseline and the skipgram performance with $p<0.0001$. The relative improvement of the average precision is 22.1 % on the simple edit distance baseline. The average precision as well as the standard deviation for each language is shown in Table VII. The average precision is compared to that of the baseline and the absolute and relative performance improvement is calculated.

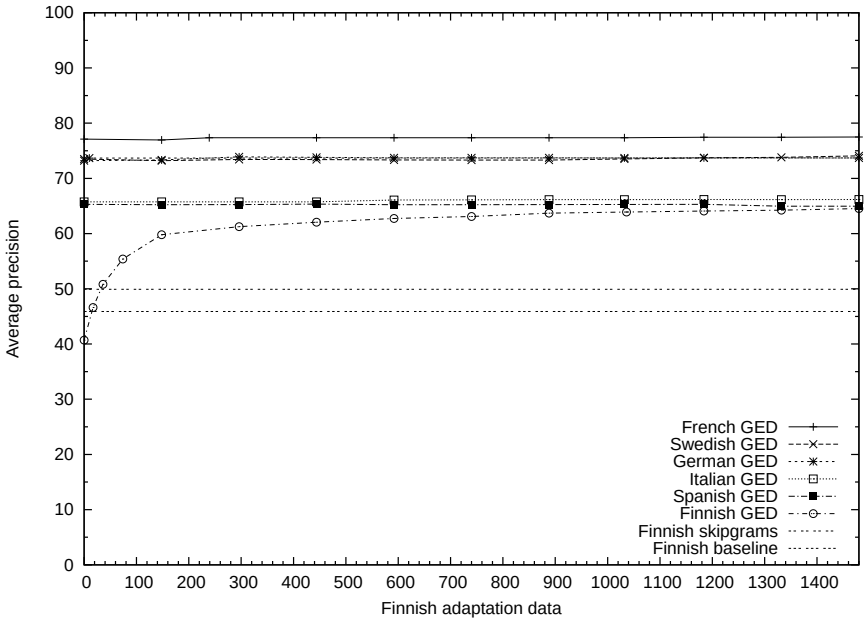

Figure 1. General edit distance (GED) performance on all the test data when adding Finnish adaptation data to the model

As can be seen in Fig. 1 and by comparing the precision of the unadapted model with the precision of the adapted model in Table VII, the performance of the other languages than Finnish did not change in any significant way while we added the Finnish adaptation data to the pool of training data. In Table VIII, we see that 64–77 % of the translations are found in the top two positions.

4.4. Importance of the Results

We have evaluated our method on terminology from several domains, and the prediction of a 26 % relative improvement of the average precision gained from the field of medical terminology was confirmed as a 22 % relative improvement on a test set composed of several domains. The improvement over the simple edit distance and skipgram baselines is statistically significant with more than 99.99 % confidence, which shows that observing the sequential ordering of the n-grams is important.

The fact that no training data was needed for Swedish and that very little adaptation data was actually needed for Finnish from a language family not present in the training data, indicates that the model has captured the essentials of transliterating technical terminology in a language independent way and even in a language-family independent way for languages using the

Table VII. The test results for each of the languages in the test data before and after the model is fully adapted to Finnish

Language	Baseline	Precision Before	Precision After	Standard Deviation	Absolute Improvmnt	Relative Improvmnt
French	72.2	77.1	77.5	±2.3	5.3	7.3
German	60.8	73.2	73.7	±2.4	12.9	21.2
Italian	53.2	65.7	66.2	±2.6	13.0	24.4
Spanish	57.0	65.3	65.0	±2.6	8.0	14.0
Finnish	45.9	40.7	64.5	±2.7	18.6	40.5
Swedish	56.0	73.5	74.1	±2.4	18.1	32.3
Average	57.5	65.9	70.2	±1.0	12.6	22.1

Table VIII. The recall percentage in positions 1, 2, 3–5, 6–10, 11–, and infinity (=not found) for each of the languages in the test data in the final model

Positions	1	2	3–5	6–10	>10	∞	Total
French	71 (198)	6 (18)	3 (11)	1 (4)	5 (14)	11 (32)	100 (277)
German	67 (193)	7 (21)	2 (6)	1 (4)	5 (16)	15 (45)	100 (285)
Italian	59 (172)	7 (21)	3 (11)	3 (9)	4 (14)	21 (62)	100 (289)
Spanish	57 (161)	7 (22)	4 (12)	3 (11)	6 (17)	20 (57)	100 (280)
Finnish	56 (154)	9 (25)	5 (14)	1 (5)	5 (15)	21 (60)	100 (273)
Swedish	66 (183)	8 (24)	3 (9)	1 (3)	5 (16)	14 (39)	100 (274)

Latin script. The fact that the performance of the other languages did not significantly change while adding Finnish adaptation data confirms that the model is robust with regard to training.

When the recall in the first two positions, 64–77 %, is compared to the average precisions, 64–78 %, at 100 % recall for the test languages, we can conclude that in practice we need only consider the top two translation candidates and most often the first candidate is the one we are looking for.

5. Discussion

In this section, we discuss the nature of cross-lingual spelling changes, especially with regard to European languages. Then we discuss the context length

and the pooling of training data in the model. Finally, we study some related work and extensions to languages with other writing systems.

5.1. Cross-lingual Spelling Changes

Most of the language independent similarity measures for finding cross-lingual spelling variants studied in CLIR, e.g., in (Keskustalo et al., 2003) and in (Pirkola et al., 2003), compare bags of n-grams. In natural language, the order of the n-grams is relevant especially within words. One of the strong points of the general edit distance is the attempt to model this ordering explicitly, e.g., the meaning of the word $blockbuster$ is not particularly close to the word $bustblocker$ despite the fact that they are only separated by four digrams: tb, kb, te, and ke. The words $blockbuster$ and $blockbusting$ are much more related even though they are separated by five digrams: te, er, ti, in, and ng.

The order of the n-grams is usually preserved when words are borrowed into another language, even if some sounds are rendered differently. In general, when we studied the location of the sound and spelling changes for the translations of medical terminology in European languages, we found that 60 % of all the changes take place in the last three letters of a word. The often very systematic nature of cross-lingual spelling variants is reflected in the fact that the word roots remain fairly intact, but the suffixes encoding a word class are different in different languages, e.g., *-tet* in Danish, *-tät* in German, *-té* in French, *-tà* in Italian, and *-ty* in English. We used this as a motivation for extending the shape of the n-gram contexts towards the end of the words in our general edit distance model.

Swedish has similar ways to Danish for productively forming new nouns and adjectives as the languages are closely related, whereas Finnish is a Fenno-Ugric language unrelated to any of the languages in the training data. Finnish, however, imposes its differing sound structure mainly on the suffixes of the technical terms. In the previous example, the corresponding Swedish and Finnish suffixes would have been *-tet* and *-teetti*, respectively. These were probably the main reasons for the good performance of Swedish and the low initial performance of Finnish.

As can be seen from the averages of the simple edit distances in the baselines for the training data, the Italian and Portuguese terminologies are the most distant ones from the English terminology. In Italian it would seem that the Latin-based terminology has in general undergone the most orthographic changes, to reflect the changing pronunciation of the words, in comparison to the other languages, where the roots of the loan words have retained more of their classical Latinate spelling.

5.2. MULTILINGUAL MODELING

The length of the n-gram context was limited by the amount of readily available training data. Using 5-character contexts would have made most of these contexts fall back on the 4-character contexts in our back-off model due to lack of training data. On the other hand, the amount of available training data for the language pairs we used was reasonable in comparison to the amount of data available for many other language pairs. The applicability of the model to other languages requires that the amount of training be kept to a minimum.

Initial experiments showed that pooling the training data gives a statistically significantly better performance than training separately for each individual language. This is probably due to the overwhelming majority of regularities present in the multilingual training data, e.g., the English term *adult* with the phonologically regular corresponding medical terms in French *adulte*, Danish *adultus*, Spanish and Portuguese *adulto* as well as Dutch *adult*[5]. These regularities compensate for irregularities in the German *erwachsen* and Italian *emancipato*. The irregularities introduce random noise in the multilingual model, which is filtered out with the frequency threshold.

The observed advantage of pooling the training data also partly explains the robustness of the model when adapting it to new languages. The model is able to gain leverage from regularities in all the training languages without being distracted by individual discrepancies. The experiment with adaptation to Finnish shows that the rendering of the sounds in the roots is not significantly different even in another language family. As soon as a small but sufficient number of regular suffix transformations have been learned, the model is able to benefit from the general sound transformations learned from other language pairs.[6]

5.3. FUTURE WORK

Some methods try to deal with spelling errors and CLSVs at the same time. However, CLSVs are fairly systematic, whereas spelling errors tend to be accidental. For a recent survey of spelling correction methods and a new method specifically designed for queries in information retrieval, see (Cucerzan and Brill, 2004). Most spelling mistakes are random insertions, deletions, changes, or transpositions of characters anywhere in a word due to mistyping. In addition, the random nature of spelling errors affects all words equally, and consequently it affects high-frequent words more often than low-frequent ones, but high-frequent words are more likely to be available in translation dictionaries. Due to this differing nature of spelling errors and CLSVs, we believe that it might be better to separate the two processes: The CLIR query is first spelling corrected based on monolingual resources, and then the query is translated filling in transliterations of new names and technical terms.

For languages with other than the Latin script, our grapheme-based model may in fact work as well as a similar model using a phoneme-based approach. Comparisons between a phoneme and grapheme-based approach in other models for transliterating English names and terminology to and from Arabic (Al-Onaizan and Knight, 2002), Japanese (Ohtake et al., 2004; Bilac and Tanaka, 2004), and Chinese (Zhang et al., 2004) support this claim. Another approach suitable for languages of different scripts is to extract explicit translations or transliterations mentioned in related corpora, e.g., (Zhang and Vines, 2004).

Another topic for further research is the impact of more advanced back-off and smoothing techniques than the ones we used. Our motivation for the way we applied Laplacian smoothing was mainly that it gave a uniform prior to the unseen events, which was the prior we used for successfully creating the training material by simple edit distance alignment. For less compatible writing systems, additional assumptions for creating the training material may be needed, and incorporating such assumptions in the back-off and smoothing techniques is probably beneficial.

6. Conclusion

We presented the problem of finding cross-lingual spelling variants in a probabilistic framework and formulated this as an instance of the general edit distance. The costs for the general edit distance were learned from a training sample of term pairs. We demonstrated that the general edit distance can be efficiently implemented with weighted finite-state transducers using context-sensitive transliterations. On the average, the top two candidates contained the intended target word more than 7 times out of 10, and approximately 2 times out of 3 the first transliteration was the one we were looking for. Our experiments also demonstrated that a distance measure, which explicitly accounts for the order of the n-grams, very significantly outperforms models based on unordered bags of n-grams. The improvement over the simple edit distance and skipgram baselines is statistically very highly significant with more than 99.99 % confidence. In addition, the experiments demonstrated that our model needed little or no adaptation data for covering new languages in the same script and that adding a new language did not significantly affect the performance of the model for the already trained languages, i.e., the model was robust under training.

In the first experiment, we trained and tested with terminology from the medical domain. Against an index of a large English newspaper database, we achieved 80–91 % precision at the point of 100 % recall for a set of medical terms in Danish, Dutch, French, German, Italian, Portuguese, and Spanish. This was a relative improvement of the average precision with 26 % on the

simple edit distance baseline. In the second experiment, we used the medical terminology as training data and tested with data consisting of terms from varied domains. We achieved 64–78 % precision at the point of 100 % recall in French, German, Italian, Spanish, Swedish, and Finnish. This is a relative improvement of 22 % on the simple edit distance baseline of the test data. For Swedish, we used no training data, and for Finnish, we needed only a small amount of training data for adapting the model.

Acknowledgments

I am grateful to Heikki Keskustalo, Kalervo Järvelin and Ari Pirkola of the University of Tampere for introducing me to the problem area and to Mathias Creutz from the Helsinki University of Technology and Lauri Carlson from Helsinki University for helpful comments on the manuscript, as well as to the anonymous reviewers. The research was done in cooperation with the Information Science Department of the Tampere University.

Notes

[1] Approximate string matching is also known as string matching allowing errors. An error model defines how different two strings are. The idea is to make the difference small when one of the strings is likely to be a variant of the other. One of the best-studied cases is the edit distance which allows deletions, insertions, and replacements of simple letters in both strings. If the different operations have different costs or the costs depend on the letters involved, we speak of the general edit distance. If all operations cost 1, we speak of simple edit distance or just edit distance. For a survey methods for approximate string matching, see (Navarro, 2001).

[2] The 50 % probability assigned to keeping a letter unchanged in an unseen context is based on a conservative intuition for European languages. The probability could be higher, but even a small amount of training material, i.e., seen contexts, will overshadow this default value.

[3] Publicly available toolkits for weighted finite-state transducers have been implemented by, e.g., (Mohri, 1997; Mohri et al., 2003) and (van Noord, 2002).

[4] The general edit distance can be efficiently implemented in the tropical semi-ring, where finding the string with the highest probability coincides with the single source shortest distance algorithms (Mohri, 2003).

[5] These are the medical terms. The popular terms for the English *adult*, i.e., grown-up, would be *volwassene* in Dutch and *voksen* in Danish, see (Stichele, 1995).

[6] Finnish may have been influenced by other European languages due to its location in Scandinavia and a comparison with, e.g., Vietnamese in its Latin script could be interesting.

References

Al-Onaizan, Y. and K. Knight: 2002, 'Machine Transliterations of Names in Arabic Text'. In: *Proceedings of ACL Workshop on Computational Approaches to Semitic Languages*.

Bilac, S. and H. Tanaka: 2004, 'A hybrid back-transliteration system for Japanese'. In: *Proceedings of the 20th International Conference on Computational Linguistics, Coling 2004*. Geneva, Switzerland, pp. 597–603.

Cucerzan, S. and E. Brill: 2004, 'Spelling correction as an iterative process that exploits the collective knowledge of web users'. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. Barcelona, Spain.

Kanji, G. K.: 1999, *100 Statistical Tests*. Sage Publications, new edition edition.

Keskustalo, H., A. Pirkola, K. Visala, E. Leppänen, and K. Järvelin: 2003, 'Non-Adjacent Digrams Improve Matching of Cross-Lingual Spelling Variants'. In: *SPIRE 2003 – 10th International Symposium on String Processing and Information Retrieval*. Manaus, Brazil.

Knight, K. and J. Graehl: 1998, 'Machine Transliteration'. *Computational Linguistics* **24**(4), 599–612.

Mohri, M.: 1997, 'Finite-State Transducers in Language and Speech Processing'. *Computational Linguistics* **23**(2), 269–311.

Mohri, M.: 2003, 'Edit-Distance of Weighted Automata'. In: J.-M. Champarnaud and D. Maurel (eds.): *Seventh International Conference, CIAA 2002*, Vol. 2608 of *Lecture Notes in Computer Science*. Tours, France, pp. 1–23, Springer, Berlin-NY.

Mohri, M., F. C. N. Pereira, and M. D. Riley: 2003, 'AT&T FSM Library – Finite-State Machine Library'. [http://www.research.att.com/sw/tools/fsm/].

Navarro, G.: 2001, 'A guided tour to approximate string matching'. *ACM Computing Surveys* **33**(1), 31–88.

Nienstedt, W.: 2003, 'Tohtori.fi – Lääkärikirja'. [http://www.tohtori.fi/laakarikirja].

Oard, D. and A. Diekema: 1998, 'Cross Language Information Retrieval'. In: *Annual Review of Information Science and Technology*, Vol. 33. pp. 223–256.

Ohtake, K., Y. Sekiguchi, and K. Yamamoto: 2004, 'Detecting Transliterated Orthographic Variants via Two Similarity Metrics'. In: *Proceedings of the 20th International Conference on Computational Linguistics, Coling 2004*. Geneva, Switzerland, pp. 709–715.

Peters, C.: 2000, 'Cross Language Evaluation Forum'. [http://clef.iei.pi.cnr.it/].

Pirkola, A., T. Hedlund, H. Keskustalo, and K. Järvelin: 2001, 'Dictionary-Based Cross-Language Information Retrieval: Problems, Methods, and Research Findings'. *Information Retrieval* **4**(3/4), 209–230.

Pirkola, A. and K. Järvelin: 2001, 'Employing the resolution power of search keys'. *Journal of the American Society of Information Science* **52**(7), 575–583.

Pirkola, A., J. Toivonen, H. Keskustalo, K. Visala, and K. Järvelin: 2003, 'Fuzzy translation of cross-lingual spelling variants'. In: *SIGIR 2003*. pp. 345–352, ACM Press.

Qu, Y., G. Grefenstette, and D. A. Evans: 2003, 'Automatic transliteration for Japanese-to-English text retrieval'. In: *SIGIR 2003*. pp. 353–360, ACM Press.

Stichele, R. V.: 1995, 'Multilingual Glossary of Technical and Popular Medical Terms in Nine European Languages'. [http://allserv.rug.ac.be/∼rvdstich/eugloss/welcome.html].

van Noord, G.: 2002, 'FSA6.2xx: Finite State Automata Utilities'. [http://odur.let.rug.nl/∼vannoord/Fsa/fsa.html].

Voutilainen, A., J. Heikkilä, and T. Järvinen: 1995, 'ENGTWOL: English Morphological Analyzer'. [http://www.lingsoft.fi/cgi-bin/engtwol].

Zhang, M., H. Li, and J. Su: 2004, 'Direct Orthographical Mapping for Machine Transliteration'. In: *Proceedings of the 20th International Conference on Computational Linguistics, Coling 2004*. Geneva, Switzerland, pp. 716–722.

Zhang, Y. and P. Vines: 2004, 'Using the web for automated translation extraction in cross-language information retrieval'. In: *SIGIR 2004*. Sheffield, United Kingdom, pp. 162–169, ACM.