

Comparison of s -gram Proximity Measures in Out-of-Vocabulary Word Translation

Anni Järvelin¹ and Antti Järvelin²

¹ `anni.jarvelin@uta.fi`

University of Tampere, Department of Information Studies, FIN-33014 University of Tampere, Finland

² `antti.jarvelin@cs.uta.fi`

University of Tampere, Department of Computer Sciences, FIN-33014 University of Tampere, Finland

Abstract. Classified s -grams have been successfully used in cross-language information retrieval (CLIR) as an approximate string matching technique for translating out-of-vocabulary (OOV) words. For example, s -grams have consistently outperformed other approximate string matching techniques, like edit distance or n -grams. The Jaccard coefficient has traditionally been used as an s -gram based string proximity measure. However, other proximity measures for s -gram matching have not been tested. In the current study the performance of seven proximity measures for classified s -grams in CLIR context was evaluated using eleven language pairs. The binary proximity measures performed generally better than their non-binary counterparts, but the difference depended mainly on the padding used with s -grams. When no padding was used, the binary and non-binary proximity measures were nearly equal, though the performance at large deteriorated.

1 Introduction

Cross-Language Information Retrieval (CLIR) refers to retrieval of documents written in a language other than that of the user's request. The document collection's language is called the *target language* and the query language the *source language*. A typical approach to CLIR is automatically translating the query into the target language. For an overview of CLIR, see [1]. Out-of-vocabulary (OOV) words constitute a major problem in query translation in CLIR. Due to the terminology missing from dictionaries, untranslatable keys appear in queries. Many typical OOV words, like proper names and technical terms, are often important query keys [2]. Therefore their translation is essential for query performance. In

© Springer-Verlag, (2008). This is the authors' version of the work. It is posted here by permission of Springer-Verlag for your personal use. The full version of this paper will be published by Springer in the proceedings of SPIRE'08 which will be published as part of Lecture Notes in Computer Science (<http://www.springer.de/comp/lncs/index.html>).

European languages, technical terms often share a common Greek or Latin root but are rendered with different spelling. This provides a good basis for the use of approximate string matching in translating the OOV words, as the words similar to a query's OOV words can be found from the target document collection and recognized as the translations of the query words.

The classified s -gram matching technique is a generalization of the well-known n -gram matching technique developed as a solution to the OOV word problem in dictionary-based CLIR [3]. In s -gram matching the strings compared are decomposed into shorter substrings, called s -grams. Skipping characters is allowed when forming the s -grams and the degree of similarity between the strings is computed by comparing their s -gram sets. s -grams, or gapped q -grams, have also been described e.g. in [4] where they were applied for fast and efficient filtering for approximate string matching. The classified s -grams differ from the other gapped q -grams in that several different s -grams are grouped together into sets of s -grams prior to calculating the similarity. The classified s -grams have been developed with CLIR and natural language processing in mind, i.e., for relatively short strings including relatively little repetition of s -grams. In CLIR applications, the technique has outperformed several other established approximate string matching techniques, such as the edit distance, the longest common subsequence and n -grams [3,5].

There are several ways of calculating the s -gram proximity between two strings. In the context of n -gram matching the L_1 distance [6], its binary version Hamming distance [7], the Dice coefficient [8], and the Jaccard coefficient [9] among others have been used. Robertson and Willett [10] mention that any proximity measure could be used, while Zobel and Dart [7] propose that measures used in IR, such as the cosine measure [8], should not be appropriate for phonetic n -gram matching as they factor out the document length.

Only similarity measures based on the Jaccard coefficient have previously been tested with classified s -grams [3,5]. Clearly, other proximity measures could also be applied, but it is not obvious which might be the best suited ones. Järvelin et al. [11] formalized a few proximity measures for s -gram matching, e.g., the L_1 distance. They argued that, theoretically, the Jaccard coefficient may not be the choice proximity measure to be used in the s -gram matching, as it is binary and thus insensitive to the counts of each s -gram in the strings to be compared. The non-binary L_1 distance should be a more sensitive proximity measure, as it takes both the types of s -grams and their number in the strings compared into account. Järvelin et al. [11] did not test their claim empirically, but their definitions enable the comparison of different s -gram proximity measures.

As the choice of the proximity measure used with the s -grams may affect the performance of the technique, testing the different proximity measures is needed. This article contributes to the issue by reporting the results of an evaluation of several proximity measures for s -gram matching of cross-lingual spelling variants. Especially the differences between the binary and the non-binary proximity measures are considered. The binary proximity measures Jaccard coefficient, binary cosine similarity and Hamming distance were compared to their

non-binary counterparts Tanimoto coefficient, cosine similarity and L_1 distance respectively. Also, the binary Dice coefficient was tested. Cross-lingual spelling variants in seven languages (English, Finnish, French, German, Italian, Spanish and Swedish) were used as source words that were translated into four target languages, English, German, Swedish and Finnish, using classified s -gram matching. In total eleven language pairs were used. The proximity measures' performance was evaluated as average translation precision.

Next, Section 2 provides an introduction to the s -grams and their proximity measures. Section 3 presents the materials and methods and Section 4 the results. Finally, section 5 contains a brief discussion and the conclusions.

2 s -gram Definitions

2.1 Introduction to s -grams

Word variation, where a language pair shares words written differently but having the same origin, is called cross-lingual spelling variation. Pirkola et al. [3] and Keskustalo et al. [5] showed that this kind of variation can advantageously be modeled with the s -grams. In s -gram matching the text strings to be compared are decomposed into substrings and the similarity between the strings is calculated as the overlap of their common substrings. Unlike in n -gram matching, skipping some characters is allowed when forming the s -grams. In CLIR applications substring length of two has been used. It has been found beneficial in IR applications to use padding spaces around the strings when forming s -grams [5,10]. This helps to get the characters at the beginning and at the end of a string properly presented in string comparison.

In *classified* s -gram matching technique [3] the s -grams originating from the same string are classified into sets based on the number of characters skipped prior to calculating the similarity. Only the s -grams belonging to the same set are compared to each other. *Gram class* indicates the skip length(s) used when generating a set of s -grams. The largest value in a gram class is called the *spanning length* of the gram class [5], e.g., for gram class $\{0, 1\}$, the spanning length is one. Two or more gram classes may also be combined into more general gram classes. The *character combination index (CCI)* then indicates the set of all the gram classes to be formed from a string, e.g. CCI $\{\{0\}, \{1, 2\}\}$ means that two gram classes are formed from a string: one with conventional n -grams formed of adjacent characters ($\{0\}$) and one with s -grams formed both by skipping one and two characters ($\{1, 2\}$). For the string "abracadabra", the s -gram set produced by the CCI $\{\{1, 2\}\}$ is $\{ar, ba, rc, aa, cd, db, bc, ra, ad, ca, ab, dr\}$, when duplicate s -grams are not listed.

2.2 s -gram Profiles and Their Proximities

s -gram-based string proximity measures are based on strings' *s-gram profiles*. The s -gram profile definitions given in this paper are extended from Ukkonen's [6]

n -gram profile definition. Next strings' s -gram profiles are defined, which are then generalized for gram classes. Then various gram class based proximity measures are given, because the strings' CCI based proximity measures are calculated as the average gram class distance of the CCI's gram classes.

Definition 1. Let $w = a_1a_2 \dots a_m$ be a string over a finite alphabet Σ , $n \in \mathbb{N}^+$ be a gram length, $k \in \mathbb{N}$ a skip length and let $x \in \Sigma^n$ be an s -gram. If $a_i a_{i+k+1} \dots a_{i+(k+1)(n-1)} = x$ for some i , then w has a $s_{n,k}$ -gram occurrence of x . Let $G_k(w)[x]$ denote the total number of $s_{n,k}$ -gram occurrences of x in w . The $s_{n,k}$ -gram profile of w is the vector $G_{n,k}(w) = (G_k(w)[x]), x \in \Sigma^n$.

s -gram profile can easily be generalized for gram classes. The gram class profiles are formed by summing up the s -gram profiles in a given gram class.

Definition 2. Let $w \in \Sigma^*$, $C \in \mathcal{P}(\mathbb{N})$ a gram class and $x \in \Sigma^n$. Let $G_C(w)[x] = \sum_{k \in C} G_k(w)[x]$. The gram class profile of w is the vector $G_{n,C}(w) = (G_C(w)[x]), x \in \Sigma^n$. In other words, $G_{n,C}(w) = \sum_{k \in C} G_{n,k}(w)$.

Sometimes the exact number of the occurrences of s -grams in the string is irrelevant, but merely the information if a specific s -gram occurs at all in the string is needed. This leads to the notion of binary gram class profile.

Definition 3. Let $w \in \Sigma^*$, and $C \in \mathcal{P}(\mathbb{N})$ a gram class and $x \in \Sigma^n$. Let

$$B_C(w)[x] = \begin{cases} 1 & \text{if } G_C(w)[x] > 0 \\ 0 & \text{otherwise} \end{cases}.$$

The binary gram class profile of w is the vector $B_{n,C}(w) = (B_C(w)[x]), x \in \Sigma^n$.

Various proximity measures can now be used to calculate string proximities based on the general and binary gram class profiles. Next, only the proximity measures using the general gram class profile of Definition 2 are given, because the corresponding proximity measures using binary profiles are defined by substituting the general s -gram profiles with binary ones in the following equations.

Let v and w be strings in Σ^* , $n \in \mathbb{N}^+$ be a gram length and $C \in \mathcal{P}(\mathbb{N})$ a gram class. L_1 distance for gram classes of strings v and w is

$$L1_{n,C}(v, w) = \sum_{x \in \Sigma^n} |G_C(v)[x] - G_C(w)[x]|. \quad (1)$$

The L_1 distance has been used with n -grams by Ukkonen [6] and its binary version, the Hamming distance, was proposed by Zobel and Dart [7]. Therefore its performance was investigated in s -gram based OOV word translation.

The cosine gram class similarity between v and w is defined as

$$Cos_{n,C}(v, w) = \frac{G_C(v)^T G_C(w)}{\|G_C(v)\| \|G_C(w)\|}, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidean norm and T the transpose of a vector. Cosine similarity (or normalized dot product) is a widely utilized proximity measure in

text retrieval applications [12] and therefore its performance in s -gram matching was also investigated along its binarized counterpart.

The Tanimoto coefficient [13] between gram classes of v and w is given by

$$T_{n,C}(v, w) = \frac{G_C(v)^T G_C(w)}{\|G_C(v)\|^2 - G_C(v)^T G_C(w) + \|G_C(w)\|^2}. \quad (3)$$

The Tanimoto coefficient was tested, because its binary counter part, the Jaccard coefficient, has traditionally been used in s -gram matching [3,5,11].

Turning to the binary profile based proximity measures, the Hamming distance $H_{n,C}(v, w)$ between v and w is derived by substituting the general gram class profile with binary profile in (1), binary cosine similarity $BinCos_{n,C}(v, w)$ by substituting with binary profiles in (2), and Jaccard coefficient $J_{n,C}(v, w)$ by doing the same substitution in (3).

Lastly, the Dice's coefficient was investigated, because it has been used in n -gram matching [10]. It is closely related to the Jaccard coefficient, but weights more the matching profile positions between the gram class profiles than the mismatching ones [12]. The Dice coefficient between v and w is given by

$$D_{n,C}(v, w) = \frac{2B_C(v)^T B_C(w)}{\|B_C(v)\|^2 + B_C(v)^T B_C(w) + \|B_C(w)\|^2}. \quad (4)$$

The character combination index based string proximity measures tested in this paper are defined as the average of strings' gram class proximities. For example, for a CCI $\mathcal{C} \in \mathcal{P}(\mathcal{P}(\mathbb{N}))$, and a gram length n , the CCI-distance corresponding to L_1 distance is

$$L1_{n,\mathcal{C}}(v, w) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} L1_{n,C}(v, w). \quad (5)$$

All CCI-based proximity measures tested below were defined analogously.

One problem that might arise when using the s -gram profiles in approximate string matching is the length of the profiles. With $s_{n,k}$ -grams, the profile length is $|\Sigma|^n$ where Σ is the specified alphabet. For example, the standard English alphabet consists of 26 letters, and thus even the di-gram profiles are quite long. However, with natural languages, the s -gram and the gram class profiles are typically very sparse, and well suited for sparse vector implementations. Therefore, the proximities between the s -gram profiles can be evaluated efficiently.

3 Materials and Methods

3.1 Materials

The test data consisted of three parts: the search keys, the target words and the set of correct translations, i.e., the relevance judgments. 271 search keys were expressed in seven languages, which were English, Finnish, French, German, Italian, Spanish and Swedish. The search keys were mostly technical terms from

the domains of biology, medicine, economics and technology, but also a list of geographical names obtained from [5] was included. These are typical cases of cross-lingual spelling variants that tend to be OOV words and thus problematic in CLIR. In total, 11 language pairs were used in the study, with four target languages: English, German, Finnish and Swedish. English was combined with all of the other languages as a target language and was also used as a source language with Finnish, German and Swedish. Translation was also done both ways between Swedish and German.

Target word lists (TWLs) consisted of CLEF 2003 [14] document collection’s indices for the target languages. The collections are full-text newspaper document collections from 1994–1995. The size of the collections, and thus the TWLs, varies between languages. The English TWL consisted of ca 257,000, the Swedish TWL of ca 388,000 and the Finnish TWL of ca 535,000 unique word forms. The German CLEF03 collection was considerably bigger and thus only a part of it was used for creating a TWL including ca 391,000 unique word forms.

All the TWLs were lemmatized (i.e. the index words were returned into their basic forms) with the TWOL morphological analyzer by Lingsoft Ltd. The words not recognized by the morphological analyzer were indexed in the word forms they appeared in the text. Compounds were split and both the original compounds and their constituents were indexed. The missing translation equivalents of the search keys were added to the TWLs, and there was only one correct translation for each search key in the TWLs.

3.2 Methods

The performance of the proximity measures was tested as follows. The *s*-gram length was set to two, because earlier research [3,5] suggests it to be the most appropriate gram length for CLIR. Padding was used at both ends of the strings and the length of the padding string was $(n - 1)(k + 1)$, where *n* is the gram and *k* the skip length. Also, *s*-grams with no padding and padding only at the beginning of strings were tested for two language pairs (English-German and German-English) with CCI $\{\{0\}, \{1, 2\}\}$ to see how the padding affects the results. For each search key 100 best translations were produced, with exception of ties in the last place when all translations within the cohort of equal proximity values were included into the result set. Translations found later were not taken into consideration, as taking more than 2-4 *s*-gram translation candidates into a query deteriorates its performance [15].

This study concentrates on comparing the proximity measures. Exhaustive testing of all possible CCIs, proximity measures and language pairs was not sensible or even possible within this study. If skip lengths 0 – 4 were considered, there would be $2^5 - 1 = 31$ possible gram classes, and thus about $2^{31} - 1$, about two billion, combinations as possible CCIs. To be able to restrict the scope of the study to some evidently useful CCIs, statistics on typical cross-lingual spelling variation between French and English and German and English were used. Pirkola et al. [16] generated statistical transformation rules that model typical character changes and correspondences between several language pairs.

Table 1. The number of transformation rules corresponding to each gram class for French to English and German to English cross-lingual spelling variants.

Gram class	{1}	{0, 1}	{1, 2}	{0, 2}	{2}	{1, 3}	{0, 3}	{2, 3}	{3}
Fr-En	56	65	44	11	12	7	3	5	3
Ge-En	117	37	36	20	7	3	4	1	0
Total	173	102	80	31	19	10	7	6	3

Table 2. The twelve CCIs used for the comparison of the proximity measures. Note that CCI₀ corresponds to the traditional n -grams. For CCI₀ gram length of two and three was experimented, for the remaining CCIs only gram length of two was used.

CCI ₀	{0}	CCI ₄	{0}, {1}	CCI ₈	{0, 1}
CCI ₁	{0}, {0, 1}	CCI ₅	{0}, {0, 1}, {1}	CCI ₉	{0, 1, 2}
CCI ₂	{0}, {0, 1}, {1}, {1, 2}	CCI ₆	{0}, {1}, {1, 2}	CCI ₁₀	{1}
CCI ₃	{0}, {0, 1}, {1, 2}	CCI ₇	{0}, {1, 2}	CCI ₁₁	{1, 2}

The rules were generated from over 10,000 term pairs of medical words. They model the same cross-lingual spelling variation phenomenon as the s -grams, but are based on an independent method and character correspondence statistics from an independent large dataset. We mapped a subset of ca 200 most frequent transformation rules to the corresponding gram classes for both language pairs and calculated the frequency of each gram class in the data.

There were some differences between the language pairs, but the transformation rules that model character changes corresponding to the gram classes {1}, {0, 1} and {1, 2} were clearly the most common ones in the data. Table 1 summarizes the results for both languages. Based on this it seemed reasonable to use only gram classes with spanning length of two or less when matching cross-lingual spelling variants. Changes corresponding to the remaining clearly less frequent gram classes were thus discarded. Keskustalo et al. [5] reached an equal conclusion, when deciding which gram classes they should use.

Based on the results of Table 1, the gram classes {1}, {0, 1}, and {1, 2} and gram class {0} corresponding to the n -grams were selected as the base gram classes for the tested CCIs. In total, the twelve CCIs of Table 2 were used in the tests. For CCI₀, in addition to the gram length of two (di-grams), also gram length of three (tri-grams) was used. This set of CCI₀ - CCI₁₁ is a representative set on effective s -grams, and by using this set a reliable picture of various s -gram proximity measures in s -gram matching can be obtained.

To compare the performances of the proximity measures, the average precision (AP, or reciprocal rank - as there is only one correct translation, these are the same) was calculated for each proximity measure at three different levels: among top 2, top 5 and top 100 highest ranked translation candidates. If the correct translation was in a cohort of words sharing the same proximity value with the target word, the average rank of the cohort was used. The top 2 and

top 5 levels were the most interesting ones, as more translation candidates would deteriorate the query performance. The Friedman test [17] was used to test the statistical significance of the differences between the proximity measures. Below, statistically significant difference corresponds to α -level $\alpha = 0.01$, statistically highly significant difference to α -level $\alpha = 0.001$, and statistically almost significant α -level $\alpha = 0.05$.

4 Results

4.1 CCIs & proximity measures over all languages

The results for all proximity measures and CCIs over all language pairs are presented in Fig. 1 and in Table 3 as the medians of AP when the top 5 translation candidates are considered. The results in top 2 and top 100 followed the same trends and are not presented due to the lack of space. The results divide the s -grams into two groups: the s -grams with CCIs that combine several s -gram types into a gram class and the s -grams where only one s -gram type is present in each gram class. In the former group (CCIs 1, 2, 3, 5, 6, 7, 8, 9, 11), the binary proximity measures performed clearly better than their non-binary counterparts, i.e., Jaccard performed better than Tanimoto, binary cosine better than cosine and Hamming better than L_1 . The differences between Jaccard and Tanimoto and binary cosine and cosine were statistically significant for 7 of these 9 CCIs for 8 language pairs out of 11. For CCI₅ the differences were statistically significant only for five language pairs (EN-FI, EN-GE, FI-EN, FR-EN, IT-EN) of which two (EN-GE, FR-EN) were only almost significant. For CCI₆ the differences were statistically significant for seven language pairs (EN-FI, EN-GE, FR-EN, GE-EN, IT-EN, SP-EN, SW-EN), two of these (IT-EN, SW-EN) being almost significant. For the two closest related language pairs (GE-SW and SW-GE) the differences were not statistically significant. Also, for EN-SW the differences were statistically significant only for CCIs 1, 2, 8, and 9. The differences between Hamming and L_1 were typically not statistically significant. The three best measures Dice, Jaccard and binary cosine performed similarly and clearly better than the rest of the proximity measures. L_1 and Hamming were the worst proximity measures. The performance difference between them and the other proximity measures was statistically significant for all language pairs and CCIs.

In the latter group, including the adjacent di-grams and tri-grams (CCI₀) and the s -grams with CCI₄ and CCI₁₀, the difference between the binary and non-binary proximity measures was smaller and always to the advantage of the non-binary measures. These differences were nevertheless never statistically significant. Tanimoto was the best proximity measure, while L_1 and Hamming were the worst ones the difference being always statistically significant. n -grams performed clearly worse than the s -grams with CCIs combining s -gram types into more general gram classes. The s -grams with CCI₄, combining two gram classes of a single s -gram type, performed better. This suggests that the s -gram technique benefits from combining gram classes into one CCI. It also seems that the

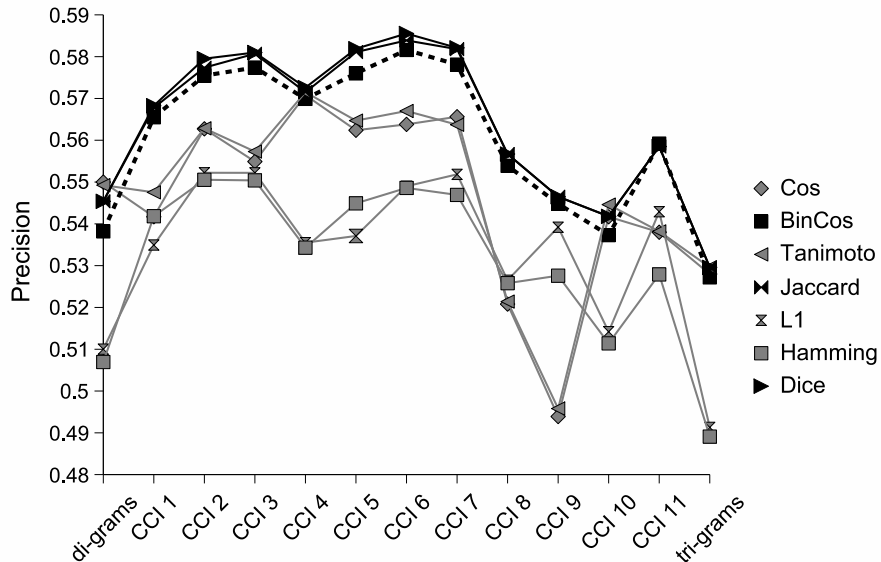


Fig. 1. The medians of APs of the proximity measures at top 5 translation candidates for all CCIs over all language pairs, zoomed in for clarity.

more s -gram types were combined into a gram class, the more the performance of Tanimoto and cosine suffered. The CCI_9 is an example of this, showing a notable fall in the performance of Tanimoto and cosine in Fig. 1.

4.2 Results for each language pair

The results in Fig. 1 and in Table 3 are medians over all the language pairs tested. To give a better picture of the results for the different language pairs, a typical CCI was selected to represent each group. CCI_6 represents the s -grams that combine several s -gram types in the gram classes. The results are presented for all language pairs in Fig. 2 as AP among the top 5 translation candidates. The binary proximity measures performed better than their non-binary counterparts. Differences between Jaccard and Tanimoto and binary cosine and cosine were statistically significant for 7 language pairs, as mentioned above (not for FI-EN, SW-GE, GE-SW, EN-SW). Dice, Jaccard and binary cosine were the best proximity measures, while L_1 and Hamming were the worst ones. The differences between these were consistently statistically significant.

Fig. 3 presents the results for CCI_0 di-grams representing the other class of s -grams as AP among the top 5 translation candidates. The results were scattered depending on the language pair, though still in line with the median results presented in Table 3 and Fig. 1. The non-binary proximity measures (Tanimoto and cosine) performed on average better than their binary counterparts, but the

Table 3. The medians of the APs of the proximity measures among top 5 translation candidates for all CCIs over all language pairs. The best proximity measures for each CCI are in bold. Tanimoto coefficient performs best for n -grams and s -grams with CCI₁₀ and Dice coefficient performs best for the s -grams with CCIs that combine several s -gram types into more general gram classes.

CCI	Proximity measure						
	Cos	BinCos	Tanimoto	Jaccard	L_1	Hamming	Dice
di-grams	0.5490	0.5382	0.5493	0.5454	0.5100	0.5070	0.5454
CCI ₁	0.5417	0.5655	0.5475	0.5678	0.5349	0.5418	0.5683
CCI ₂	0.5627	0.5755	0.5629	0.5774	0.5522	0.5506	0.5795
CCI ₃	0.5549	0.5774	0.5573	0.5807	0.5522	0.5504	0.5810
CCI ₄	0.5708	0.5699	0.5715	0.5715	0.5355	0.5343	0.5726
CCI ₅	0.5624	0.5760	0.5647	0.5811	0.5371	0.5449	0.5819
CCI ₆	0.5638	0.5816	0.5670	0.5839	0.5490	0.5486	0.5855
CCI ₇	0.5656	0.5781	0.5637	0.5818	0.5518	0.5469	0.5821
CCI ₈	0.5208	0.5539	0.5214	0.5567	0.5265	0.5258	0.5567
CCI ₉	0.4939	0.5448	0.4958	0.5465	0.5392	0.5276	0.5465
CCI ₁₀	0.5417	0.5373	0.5446	0.5418	0.5142	0.5114	0.5418
CCI ₁₁	0.5380	0.5592	0.5382	0.5585	0.5429	0.5279	0.5585
tri-grams	0.5280	0.5272	0.5296	0.5296	0.4913	0.4891	0.5296
MEDIAN	0.5490	0.5655	0.5493	0.5678	0.5371	0.5343	0.5683

differences were not statistically significant. Hamming distance and L_1 were the worst measures, with statistically significant difference to the other proximity measures. Tri-grams performed generally worse than di-grams.

4.3 Padding

The differences between the binary and non-binary proximity measures were clearly reduced when no padding or padding only at the beginning of the strings were used. When no padding at all was used, the results deteriorated for all proximity measures and more for the binary than the non-binary proximity measures. For cosine and Tanimoto, the results even improved slightly for one of the two language pairs (GE-EN). Thus the differences between corresponding binary and non-binary proximity measures were reduced and were not statistically significant. When only the left-side padding was used, the overall effect on results was a little unclear: for English to German matching the best results deteriorated slightly, but for German to English the top results improved slightly. The non-binary proximity measures improved in comparison to their binary counterparts and the differences between them were not statistically significant. L_1 and Hamming suffered both from not using padding and also from using padding only at the beginning of the strings. They were always clearly the worst proximity measures with a statistically highly significant difference between them and the other proximity measures.

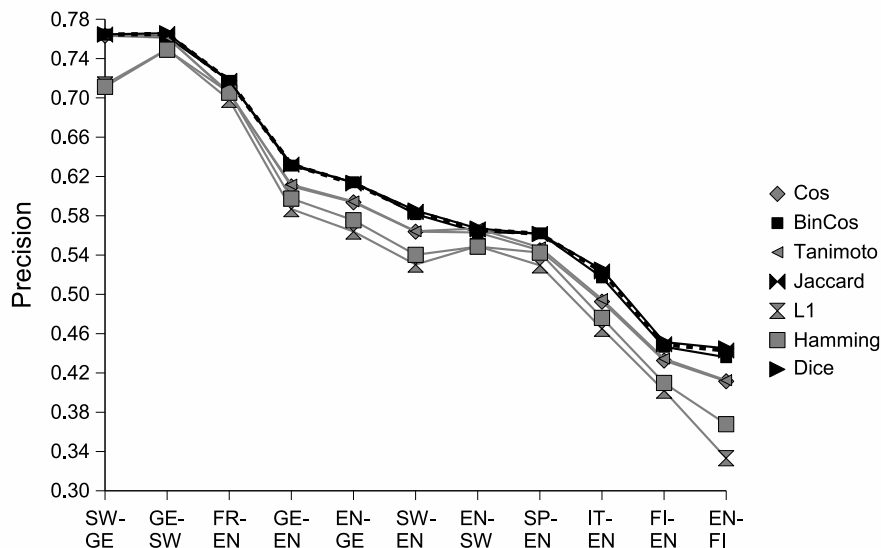


Fig. 2. The AP of the proximity measures at top 5 for all language pairs for the s -grams with CCI_6 . The figure is zoomed in for clarity.

5 Discussion

To sum up the results, the binary proximity measures performed better than their non-binary counterparts in s -gram based matching of OOV words. Dice, Jaccard and binary cosine performed best and any of these measures could be beneficially used. The difference between the binary and non-binary proximity measures seems to depend on the CCI used: when a number of different s -gram types were combined into a more general gram class (such as $\{\{1, 2\}\}$), the binary proximity measures clearly outperformed their non-binary counterparts. For the CCIs where only one s -gram type was present in each gram class (the traditional n -grams, CCI_4 , and CCI_{10}), the differences between the binary and non-binary proximity measures vanished. Also, the more s -gram types were combined into a gram class, the more the performance of Tanimoto and cosine suffered.

This seems to be linked to the padding used with s -grams: When several s -gram types are combined into one gram class and padding was used, identical s -grams from both ends of strings are formed repeatedly and become overweighted when using non-binary proximity measures. As character changes are especially common at the ends of cross-lingual spelling variants (e.g. *antiseptic* - *antiseptique*), this damages the performance of the non-binary proximity measures. Removing the padding is nevertheless not a guarantee of success as it may affect the overall performance of the s -gram matching negatively. Keskustalo et al. [5] have found earlier that whether the padding on both sides of strings or only at the beginning performs best depends on the language pair at hand. For s -

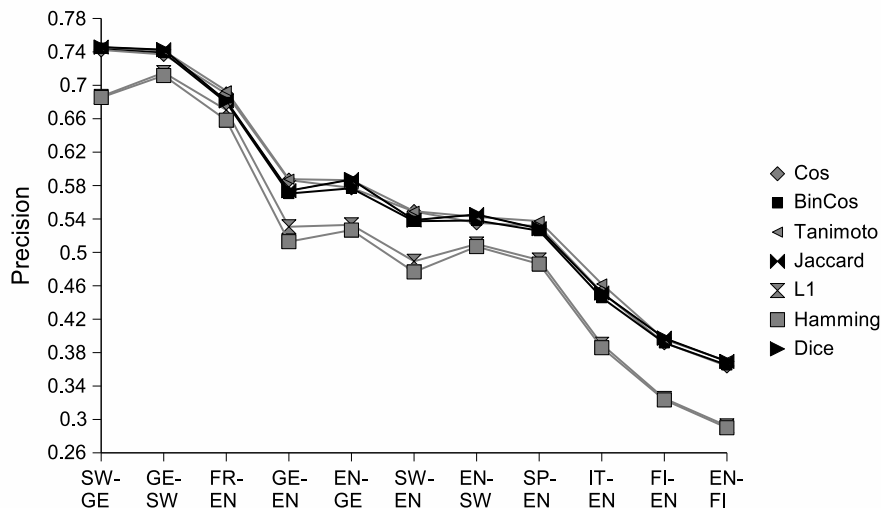


Fig. 3. The AP of the proximity measures at top 5 for all language pairs for traditional di-grams (CCI_0). The figure is zoomed in for clarity.

gram matching implementations using non-binary s -gram profiles, the repetitive occurrences of s -grams including padding characters should be ignored.

L_1 and its binary counterpart Hamming distance did not perform well and they do not seem suitable proximity measures for this application area. With these proximity measures the distance between two strings is calculated as the mean value of the different s -grams in the gram classes. This causes the measures to favor short words as no s -grams can be formed of one letter words (without padding) and none or very few of two or three letter words. Therefore, L_1 and Hamming give more non-relevant short words at the top ranks in the result lists than the other proximity measures. This is also reflected in the fact that the results for L_1 and Hamming deteriorated when the padding was removed.

Non-binary proximity measures are suitable for applications where a lot of repetition of s -grams occur (e.g. gene matching). In cross-lingual OOV word matching the alphabet used is rather large and the strings processed quite short. Consequently the repetition of s -grams is not extensive and therefore the binary and non-binary s -gram profiles approach each other. Therefore, no advantage is achieved with the use of the non-binary proximity measures.

Acknowledgments.

The authors wish to thank Academy Professor Kalervo Järvelin, Ph.D., Docent Ari Pirkola, Ph.D., and Mr. Heikki Keskustalo, M.Sc. from University of Tampere for their support and comments on the paper. The first author was funded by Tampere Graduate School in Information Science and Engineering (TISE) and Academy of Finland under grant # 1209960.

References

1. Kishida, K.: Technical issues of cross-language information retrieval: a review. *Inf. Process. Manage* **41**(3) (2005) 433–455
2. Pirkola, A., Järvelin, K.: Employing the resolution power of search keys. *JASIST* **52**(7) (2001) 575–583
3. Pirkola, A., Keskustalo, H., Leppänen, E., Käsälä, A.P., Järvelin, K.: Targeted s-gram matching: a novel n-gram matching technique for cross- and monolingual word form variants. *Information Research* **7**(2) (2002) Available at <http://InformationR.net/ir/7-2/paper126.html>.
4. Burkhardt, S., Kärkkäinen, J.: Better filtering with gapped q-grams. *Fundamenta Informaticae* **56**(1–2) (2003) 51–70
5. Keskustalo, H., Pirkola, A., Visala, K., Leppänen, E., Järvelin, K.: Non-adjacent digrams improve matching of cross-lingual spelling variants. In: *Proceedings of the 10th International Symposium on String Processing and Information Retrieval (SPIRE)*. Volume 2857 of LNCS., Berlin, Germany, Springer (2003) 252–265
6. Ukkonen, E.: Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci.* **92**(1) (1992) 191–211
7. Zobel, J., Dart, P.: Phonetic string matching: lessons from information retrieval. In: *SIGIR '96: Proceedings of the 19th ACM SIGIR Conference*, New York, NY, USA, ACM Press (1996) 166–172
8. Salton, G., McGill, M.J.: *Introduction to Modern Information Retrieval*. 1st edn. McGraw-Hill, New York, NY, USA (1983)
9. Pfeiffer, U., Poersch, T., Fuhr, N.: Retrieval effectiveness of proper name search methods. *Inf. Process. Manage* **32**(6) (1996) 667–679
10. Robertson, A.M., Willett, P.: Applications of n-grams in textual information systems. *J Doc* **54**(1) (1998) 48–69
11. Järvelin, A., Järvelin, A., Järvelin, K.: s-grams: defining generalized n-grams for information retrieval. *Inf. Process. Manage* **43**(4) (2007) 1005–1019
12. Hand, D.J., Mannila, H., Smyth, P.: *Principles of Data Mining*. 1st edn. MIT Press, Cambridge, MA, USA (2001)
13. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*. 2nd edn. Academic Press, London, UK (2003)
14. Peters, C.: *Introduction to the CLEF 2003 working notes* (2003) Available at: <http://clef.iei.pi.cnr.it/>.
15. Hedlund, T., Airio, E., Keskustalo, H., Lehtokangas, R., Pirkola, A., Järvelin, K.: Dictionary-based cross-language information retrieval: Learning experiences from CLEF 2000 – 2002. *Information Retrieval - Special Issue on CLEF Cross-Language IR* **7**(1–2) (2004) 99–119
16. Pirkola, A., Toivonen, J., Keskustalo, H., Visala, K., Järvelin, K.: Fuzzy translation of cross-lingual spelling variants. In: *SIGIR '03: Proceedings of the 26th ACM SIGIR Conference*, New York, NY, USA, ACM Press (2003) 345–352
17. Conover, W.J.: *Practical Nonparametric Statistics*. 3rd edn. Wiley, New York, NY, USA (1999)