Character *N*-Grams Translation in Cross-Language Information Retrieval

Jesús Vilares[1], Michael P. Oakes[2], and Manuel Vilares[3]

[1] Department of Computer Science, University of A Coruña
Campus de Elviña s/n, 15071 – A Coruña (Spain)
jvilares@udc.es
[2] School of Computing and Technology, University of Sunderland
St. Peter's Campus, St. Peter's Way, Sunderland – SR6 0DD (United Kingdom)
Michael.Oakes@sunderland.ac.uk
[3] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 – Ourense (Spain)
vilares@uvigo.es

Abstract. This paper describes a new technique for the direct translation of character n-grams for use in Cross-Language Information Retrieval systems. This solution avoids the need for word normalization during indexing or translation, and it can also deal with out-of-vocabulary words. This knowledge-light approach does not rely on language-specific processing, and it can be used with languages of very different natures even when linguistic information and resources are scarce or unavailable. Our proposal also tries to achieve a higher speed during the n-gram alignment process with respect to previous approaches.

Key words: Cross-Language Information Retrieval, character n-grams, translation algorithms, alignment algorithms, association measures.

1 Introduction

The interest in using character n-grams for text conflation in Information Retrieval (IR) comes from the possibilities they offer, particularly in the case of non-English languages [6, 7]. Since it provides a surrogate means to normalize word forms and it does not rely on language-specific processing, it can be applied to very different languages, even when linguistic information and resources are scarce or unavailable.

Its use is quite simple, since both queries and documents are just tokenized into their compounding overlapping n-grams instead of words: the word `potato`, for example, is split into: `-pot-`, `-ota-`, `-tat-` and `-ato-`. The resulting n-grams are then processed by the retrieval engine.

Nevertheless, when extending its use to Cross-Language Information Retrieval (CLIR), an extra translation phase is needed. A simple solution consists of, firstly, using any of the standard machine translation methods used in CLIR for translating the query and, next, splitting the resulting query into n-grams [6].

Our approach is based on the previous work of the Johns Hopkins University Applied Physics Lab (JHU/APL), which went one step further and proposed a direct n-gram translation algorithm which allowed translation not at the word level but at the n-gram level [7]. This solution avoids some of the limitations of classic dictionary-based translation methods, such as the need for word normalization or the inability to handle out-of-vocabulary words. Nevertheless, the initial proposal resulted to be very slow. For example, it could take several days in the case of working with 5-grams.

This paper describes a new proposal for direct n-gram translation we have developed and which tries to speed up the process in order to make the testing of new developments easier. The article is structured as follows. Firstly, Sect. 2 describes our system. Next, Sect. 3 evaluates our approach. Finally, Sect. 4 presents our conclusions and future work.

2 The Character *N*-Gram Alignment Algorithm

In contrast with the original system developed by JHU/APL, which relies mainly on ad-hoc resources, our system has been built using freely available resources when possible in order to minimize effort and to make it more transparent. This way, our system employes the open-source retrieval platform Terrier [1]. This decision was supported by the satisfactory results obtained with n-grams using different indexing engines [11]. The well-known Europarl parallel corpus [3] is also used. This corpus was extracted from the proceedings of the European Parliament, containing up to 28 million words per language. It includes versions in 11 European languages: Romance (French, Italian, Spanish, Portuguese), Germanic (English, Dutch, German, Danish, Swedish), Greek and Finnish.

Our n-gram alignment algorithm consists of two phases. In the first phase, the slowest one, the input parallel corpus is aligned at the word-level using the well-known statistical tool GIZA++ [9], obtaining as output the translation probabilities between the different source and target language words. Next, in the second phase, n-gram translation scores are computed employing statistical association measures [5]. Our approach increases the speed of the process by concentrating most of the complexity in the word-level alignment phase. This first step acts as a filter, since only those n-gram pairs corresponding to aligned words will be considered, whereas in the original JHU/APL approach all n-gram pairs corresponding to aligned paragraphs were considered.

2.1 Word-Level Alignment Using Association Measures

Our n-gram alignment algorithm is an extension of the way association measures can be used for creating bilingual word dictionaries taking as input parallel collections aligned at the paragraph level [12]. In this context, given a word pair $(word_s, word_t)$ —$word_s$ standing for the source language word, and $word_t$ for its candidate target language translation—, their cooccurrence frequency can be organized in a *contingency table* resulting from a cross-classification of their cooccurrences in the aligned corpus:

	$T = word_t$	$T \neq word_t$	
$S = word_s$	O_{11}	O_{12}	$= R_1$
$S \neq word_s$	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

As shown, the first row accounts for those instances where the source language paragraph contains $word_s$, while the first column accounts for those instances where the target language paragraph contains $word_t$. The cell counts are called the *observed frequencies*: O_{11}, for example, stands for the number of aligned paragraphs where the source language paragraph contains $word_s$ and the target language paragraph contains $word_t$. The total number of word pairs considered —or *sample size* N— is the sum of the observed frequencies. The row totals, R_1 and R_2, and the column totals, C_1 and C_2, are also called *marginal frequencies*, and O_{11} is called the *joint frequency*.

Once the contingency table has been built, different association measures can be easily calculated for each word pair. The most promising pairs, those with the highest association measures, are stored in the bilingual dictionary.

2.2 Adaptations for *N*-Gram-Level Alignment

We have described how to compute and use association measures for generating bilingual word dictionaries from parallel corpora. However, our context is different, since we do not have aligned paragraphs composed of words, but aligned words —previously aligned through GIZA++— composed of n-grams. A first choice could be just to adapt the contingency table to this context, by considering that we are managing n-gram pairs $(n\text{-}gram_s, n\text{-}gram_t)$ cooccurring in aligned words instead of word pairs $(word_s, word_t)$ cooccurring in aligned paragraphs. So, contingency tables should be adapted accordingly: O_{11}, for example, should be re-formulated as the number of aligned word pairs where the source language word contains $n\text{-}gram_s$ and the target language word contains $n\text{-}gram_t$.

This solution seems logical, but is not completely accurate. In the case of aligned paragraphs, we had *real* instances of word cooccurrences at the paragraphs aligned. However, now we do not have *real* instances of n-gram cooccurrences at aligned words, but just *probable* ones, since GIZA++ uses a statistical alignment model which computes a translation probability for each cooccurring word pair [9]. So, the same word may be aligned with several translation candidates, each one with a given probability. Taking as example the case of the English words `milk` and `milky`, and the Spanish words `leche` *(`milk`)*, `lechoso` *(`milky`)* and `tomate` *(`tomato`)*, a possible output word-level alignment would be:

source word	**candidate translation**	**probability**
`milk`	`leche`	0.98
`milky`	`lechoso`	0.92
`milk`	`tomate`	0.15

This way, it may be considered that the source 4-gram `-milk-` does not *really* cooccur with the target 4-gram `-lech-`, since the alignment between its containing words `milk` and `leche`, and `milky` and `lechoso` is not certain. Nevertheless,

it seems much more probable that the "*translation*" of `-milk-` is `-lech-` rather than `-toma-`, since the probability of the alignment of their containing words —`milk` and `tomate`— is much smaller than that of the words containing `-milk-` and `-lech-` —the pairs `milk` and `leche` and `milky` and `lechoso`. Taking this idea as a basis, our proposal consists of weighting the likelihood of a cooccurrence according to the probability of its containing alignments.

So, the resulting contingency tables corresponding to the n-gram pairs *(`-milk-`, `-lech-`)* and *(`-milk-`, `-toma-`)* are as follows:

	$T =$ `-lech-`	$T \neq$ `-lech-`	
$S =$ `-milk-`	$O_{11} = 0.98 + 0.92 =$**1.90**	$O_{12} = 0.98 + 3 * 0.92 + 3 * 0.15 =$**4.19**	$R_1 =$**6.09**
$S \neq$ `-milk-`	$O_{21} =$**0.92**	$O_{22} = 3 * 0.92 =$**2.76**	$R_2 =$**3.68**
	$C_1 =$**2.82**	$C_2 =$**6.95**	$N =$**9.77**

	$T =$ `-toma-`	$T \neq$ `-toma-`	
$S =$ `-milk-`	$O_{11} =$**0.15**	$O_{12} = 2 * 0.98 + 4 * 0.92 + 2 * 0.15 =$**5.94**	$R_1 =$**6.09**
$S \neq$ `-milk-`	$O_{21} =$**0**	$O_{22} = 4 * 0.92 =$**3.68**	$R_2 =$**3.68**
	$C_1 =$**0.15**	$C_2 =$**9.62**	$N =$**9.77**

Notice that, for example, the O_{11} frequency corresponding to *(`-milk-`, `-lech-`)* is not 2 as might be expected, but 1.90. This is because the pair appears in two alignments, `milk` with `leche` and `milky` with `lechoso`, but each cooccurrence in an alignment has been weighted according to its translation probability:

$$O_{11} = 0.98 \text{ (for } \texttt{milk} \text{ with } \texttt{leche}) + 0.92 \text{ (for } \texttt{milky} \text{ with } \texttt{lechoso}) = 1.90 \ .$$

Once the contingency tables have been generated, the association measures can be computed. Our system employes two classic measures: the *Dice coefficient* (*Dice*) and *mutual information* (*MI*), defined by the following equations [5]:

$$Dice(n\text{-}gram_s, n\text{-}gram_t) = \frac{2O_{11}}{R_1 + C_1}. \quad (1) \qquad MI(n\text{-}gram_s, n\text{-}gram_t) = log\frac{NO_{11}}{R_1 C_1}. \quad (2)$$

If using the Dice coefficient, for example, we find that the association measure of the pair *(`-milk-`, `-lech-`)* —the correct one— is much higher than that of the pair *(`-milk-`, `-toma-`)* —the wrong one:

$Dice$(`-milk-`, `-lech-`)$= \frac{2*1.90}{6.09+2.82} =$ **0.43** . $Dice$(`-milk-`, `-toma-`)$= \frac{2*0.15}{6.09+0.15} =$ **0.05** .

3 Evaluation

Before trying with less well-known languages with a greater lack of resources —which are the aim of this approach—, our system has to be tuned and studied more in depth. For this purpose, our approach has been initially tested in English-to-Spanish bilingual runs using the English topics and the Spanish document collections of the CLEF 2006 *robust task* [8]. The Spanish data collection is formed by 454,045 news reports (1.06 GB), while the test set consists of the

60 topics (C050–C059, C070–C079, C100–C109, C120–C129, C150–159, C180–189) of the *training topics* subset established for that task. Topics are formed by three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria. Nevertheless, only *title* and *description* fields have been used, simulating in this way the case of "*short*" queries as those used in commercial engines [8].

Regarding the indexing process, documents were lowercased and punctuation marks —but not diacritics— were removed. Finally, the texts were split into n-grams and indexed, using 4-grams as a compromise n-gram size after studying the previous results of the JHU/APL group [7]. The open-source TERRIER platform [1] has been employed as the retrieval engine, using a InL2[4] ranking model [2]. No stopword removal or query expansion were applied at this point.

For querying, the source language topic is firstly split into n-grams. Next, these n-grams are replaced by their candidate translations according to a selection algorithm, and the resulting translated topics are then submitted to the retrieval system. Two selection algorithms are currently available: a *top-rank-based* algorithm, that takes the N highest ranked n-gram alignments according to their association measure, and a *threshold-based* algorithm, that takes those alignments whose association measure is greater or equal than a threshold T.

Next, we present the results obtained with the association measures currently implemented in our system: the Dice coefficient and mutual information.[5]

3.1 Results Using the Dice Coefficient

Results Using Unidirectional Word-Level Alignment. Our first tests with the Dice coefficient used the top-rank-based selection algorithm, that is, by taking the target n-grams from the N top n-gram-level alignments with the highest association measures.[6] The best results were obtained when using a limited number of translations, those with N=1 being the best ones. Such results are displayed in the left-hand Precision vs. Recall graph of Fig. 1, labeled as '`W=0.00 N=1`' —notice that mean average precision (MAP) values are also given.

The next tests used the threshold-based selection algorithm, that is, by fixing a minimal association measure threshold T.[7] The best run, using T=0.30, is shown in the left-hand graph of Fig. 1 labeled as '`W=0.00 T=0.30`'. As can be seen, the results obtained were significantly less good as the previous ones.[8]

Next, trying to reduce the noise introduced in the system by word-level translation ambiguities and, in this way, to improve the n-gram alignment, we removed from the input those least-probable word alignments. After studying the distribution of the input aligned word pairs across their translation probabilities, we

[4] Inverse Document Frequency model with Laplace after-effect and normalization 2.

[5] These experiments must be considered as *unofficial* experiments, since the results obtained have not been checked by the CLEF organization.

[6] With $N \in \{1, 2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$.

[7] With $T \in \{0.00, 0.001, 0.01, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$.

[8] Two-tailed T-tests over MAPs with α=0.05 have been used along this work.

Table 1. General distribution of input aligned word pairs across their translation probabilities.

	unidir. alignment		*bidir. alignment*	
	W=0.00	*W=0.15*	*W=0.00*	*W=0.15*
#pairs	2,155,482	66,610	672,502	32,011
μ	0.0233	0.2936	0.0287	0.3489
σ	0.0644	0.1845	0.0887	0.2116

Table 2. General distribution of output aligned n-gram pairs across their association measures: the Dice coefficient and mutual information

		unidir. alignment		*bidir. alignment*	
		W=0.00	*W=0.15*	*W=0.00*	*W=0.15*
#pairs		18,463,772	1,166,930	6,828,044	600,120
Dice	μ	0.0036	0.0644	0.0133	0.1439
	σ	0.0261	0.1355	0.0721	0.2252
MI	μ	-0.6672	4.3056	-0.1476	5.2094
	σ	3.8994	2.3019	4.0581	2.4206

decided to dismiss those pairs with a probability less than a threshold W=0.15. This way we reduced the number of input pairs processed by 97%, from 2,155,482 to 66,610 —see Table 1—, and by 94% the number of output n-gram pairs generated, from 18,463,772 to 1,166,930 —see Table 2. This resulted in a considerable reduction of processing and storage resources, processing time included.

On the other hand, according to Tables 1 and 3, the level of ambiguity was reduced in both the input and output. In the case of the input, the mean number of possible translations per source word in the input word-level alignment was reduced from 41.1477 translations per source word with a mean probability of 0.0233, to 2.0049 translations with a mean probability of 0.2936. This implies a

Table 3. General distribution of source-language terms across their number of possible translations: in the input aligned word pairs (*left*), and in the output aligned n-gram pairs (*right*).

	input aligned word pairs				*output aligned n-gram pairs*			
	unidir. alignment		*bidir. alignment*		*unidir. alignment*		*bidir. alignment*	
	W=0.00	*W=0.15*	*W=0.00*	*W=0.15*	*W=0.00*	*W=0.15*	*W=0.00*	*W=0.15*
#terms	52,384	33,223	48,935	28,238	35,728	30,880	33,818	27,932
μ	41.1477	2.0049	13.7427	1.1336	516.7871	37.7892	201.9056	21.4850
σ	76.1284	1.4717	43.1740	0.3858	949.8868	82.6615	502.7873	50.0478

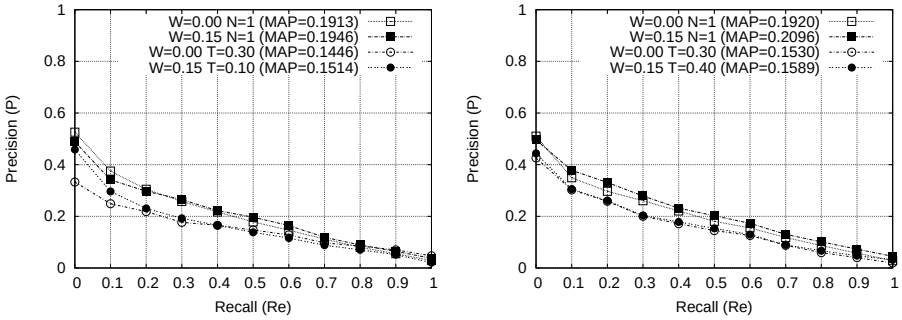

Fig. 1. Precision vs. Recall graphs of the test runs performed using the Dice coefficient and taking as input a unidirectional (*left*) or bidirectional (*right*) word-level alignment.

reduction of 95% in the number of possible translations and a parallel increase of 1160% in their mean translation probability.

In the case of the output, according to Tables 2 and 3, the mean number of possible translations per source n-gram in the output was reduced from 516.7871 translations with a mean association measure of 0.0036, to 37.7892 translations with a mean measure of 0.0644. This implies a reduction of 93% in the number of translations and a increase of 1689% in their association measure.

The results obtained introducing this refinement are no significantly different, in general, from those obtained without pruning, whatever the selection algorithm used. Those best results obtained for each selection approach —with N=1 and T=0.10— are shown in the left-hand graph of Fig. 1. As can be seen, the top-rank-based selection algorithm keeps performing significantly better.

So, we can conclude that although this refinement does not really improve the results, it reduces considerably those computing and storage resources required by the system, justifying its application. On the other hand, the system showed to be robust against the noise introduced by the high percentage of low-probability alignments of the input.

Results Using Bidirectional Word-Level Alignment. Once more, we tried to reduce the noise introduced in the system, this time by refining the initial word-level alignment by using a bidirectional alignment [4]. That is, we considered a $(word_{English}, word_{Spanish})$ English-to-Spanish alignment only if there also existed a corresponding $(word_{Spanish}, word_{English})$ Spanish-to-English alignment. This way we focus the processing on those words whose translation seems less ambiguous. The best results obtained for this approach are presented in the right-hand graph of Fig. 1. We will discuss now the impact of this refinement, taking as the baseline those runs obtained using a unidirectional algorithm where no minimal word-level translation probability threshold was fixed —i.e., W=0.

By examining Table 1 we can see that the bidirectional alignment reduced the number of input word pairs by 69% —from 2,155,482 to 672,502 pairs— and, according to Table 2, it reduced the number of output n-gram pairs by 63%

—from 18,463,772 to 6,828,044 pairs. This reductions allows us to reduce both computing and storage resources —including processing time.

Regarding the level of ambiguity, in the case of the input, Tables 1 and 3 show a reduction from 41.1477 translations per input source word with a mean probability of 0.0233, to 13.7427 translations with a probability of 0.0287. This means a reduction of 67% in the number of translations and a increase of 23% in the translation probability of the input. In the case of the output, Tables 2 and 3 show a reduction from a mean of 516.7871 translations per source n-gram with a mean association measure of 0.0036, to 201.9056 translations with a measure of 0.0133; a reduction of 61% and a increment of 269%, respectively.

With respect to the results, the best ones, obtained again with N=1 and T=0.30, are shown in the right-hand graph of Fig. 1, being not significantly different from those obtained with the original unidirectional alignment, with the top-rank-based selection algorithm performing significantly better than the threshold-based approach. So, we can conclude that the use of bilingual alignment does not damage the performance of the system, and also reduces computing and storage resources —including processing time. The system was also demonstrated to be robust against inaccurate or ambiguous input alignments.

We have also considered combining the word-level bilingual alignment with the use of the word-level translation probability threshold W, looking for an extra reduction of both the level of ambiguity and the computing and storage resources needed. Taking as the baseline the results obtained when applying such a probability threshold W=0.15 over the original unidirectional alignment, Table 1 shows an extra 52% reduction —from 66,610 to 32,011 pairs— in the number of input word alignments, and an extra 49% reduction —from 1,166,930 to 600,120 pairs— in the output n-gram alignments.

With respect to the level of ambiguity, there is an extra 43% reduction —from 2.0049 to 1.1336 pairs— in the mean number of input word translations, with an 19% increment —from 0.2936 to 0.3489— of the mean word translation probability. In the case of the output n-gram translations, their mean number of translations was reduced from 37.7892 to 21.4850 pairs (43%), with a increase of the mean association measure from 0.0644 to 0.1439 (123%).

The results obtained, shown in the right-hand graph of Fig. 1, continue being not significantly different from the initial ones, with the top-rank-based selection algorithm performing significantly better. On the other hand, they show no apparent damage to the performance, allowing us to conclude that the combined use of both refinements minimizes the resources required by the system without harming its performance.

3.2 Results Using Mutual Information

Our second main set of experiments used mutual information (MI) as the association measure. The main difference with respect to the Dice coefficient is that the Dice coefficient takes values within the range $[0\,..\,1]$, while MI can take any value within $(-\infty\,..\,+\infty)$. Moreover, negative MI values correspond to pairs of

terms avoiding each other, while positive values point out cooccurring terms. Finally, MI also tends to overestimate low-frequency data.

These features had to be taken into account in order to adapt our testing methodology. In the case of the top-rank-based selection algorithm, we continued taking the N top-ranked n-gram alignments, even if their MI value was negative. However, in the case of the threshold-based algorithm, since the range of MI values for each test run may vary considerably, the threshold values were fixed according to the following formula in order to homogenize the tests:

$$T_i = \mu + 0.5\ i\ \sigma\ . \tag{3}$$

where T_i represents the i-th threshold, with $i \in \mathbb{N}_0$, μ represents the *mean* of the MI values obtained for the present configuration, and σ its *standard deviation*. This way, the first threshold was fixed at $T_0 = \mu$, the following threshold at $T_1 = \mu + 0.5\ \sigma$, next at $T_1 = \mu + \sigma$, and so on, until reaching the highest possible threshold without overpassing the maximal MI value for the present configuration.

Results Using a Unidirectional Word-Level Alignment. This first test run corresponds to a unidirectional alignment using the top-rank-based selection algorithm with no word-level pruning —i.e., W=0.00. Results were not as good as those obtained using the Dice coefficient. The best run, that one using N=30, is presented in the left-hand graph of Fig. 2.

When introducing the word-level translation probability threshold W=0.15, the gains were the same as with the Dice coefficient, except for the mean association measure. This is because word-level gains —reduction of input word pairs and increment of the mean translation probability— only depend on the value of W, and are not affected by the association measure. At the n-gram level, the reduction in the number of output n-gram pairs only depends on the input word pairs —and, consequently, on W. Nevertheless, the mean association measures will vary, since we are now using MI. Mean values are shown in Table 2, and we can see how they increased from -0.6672 to +4.3056 (745%).

The results obtained were not significantly different from those obtained with W=0.00. The best ones, those for N=20, are shown in the left-hand graph of Fig. 2. As in the case of the Dice coefficient, the introduction of the threshold W does not damage the performance of the system, but reduces the computing and storage resources required. On the other hand, the system demonstrated again its robustness against the distortion introduced by low-probability inputs.

When using the threshold-based algorithm, results were slightly better than those with the top-rank-based algorithm —except at the lowest recall levels—, although this difference was not significant. Results improved when raising the threshold, but continued being not as good as those obtained with the Dice coefficient. The results for the best run, with $T = \mu + 2.5\ \sigma$, are shown in the left-hand graph of Fig. 2.

When pruning the input data by applying the word-level probability threshold W=0.15, the results seemed to approach even more those obtained with the

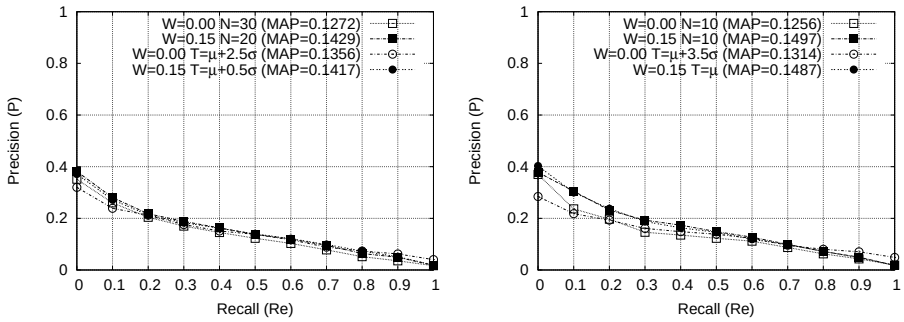

Fig. 2. Precision vs. Recall graphs of the test runs performed using mutual information and taking as input a unidirectional (*left*) or bidirectional (*right*) word-level alignment.

top-rank-based algorithm. As before, no significant difference was found with respect to the results obtained without pruning. In this case the best threshold was $T = \mu + 0.5\ \sigma$, as shown in the left-hand graph of Fig. 2.

Results Using a Bidirectional Word-Level Alignment. Our last set of test runs introduced again a word-level bidirectional alignment. The results obtained when using the top-rank-based selection algorithm were not significantly different from those obtained when employing a unidirectional alignment, whether we use W=0.00 or W=0.15 —see right-hand graph of Fig. 2.

As before, the gains obtained with the word-level threshold W were the same as with the Dice coefficient, except for the mean association measure. When taking as the baseline the unidirectional run with W=0.15, Table 2 shows an 21% increment of the mean MI value, from 4.3056 to 5.2094.

In the case of using a threshold-based selection algorithm, the results obtained were again not significantly different from those obtained with an unidirectional alignment, as shown in the right-hand graph of Fig. 2.

So, we can conclude that, as with the Dice coefficient, the introduction of a bidirectional alignment does not damage the performance of the system, but reduces the resources required. On the other hand, the system showed again its robustness against inaccurate or ambiguous input word alignments.

Finally, to complete this evaluation section, Fig. 3 shows the best results obtained for each combination of association measure and word-level alignment approach, with respect to several baselines: by querying the Spanish index with the English topics split into 4-grams (`EN 4-grams`) —allowing us to measure the impact of casual matches—, by querying the Spanish index using the stemmed Spanish topics[9] (`ES stemming`), and by querying the Spanish index using the Spanish

[9] We have used the Snowball stemmer (`http://snowball.tartarus.org`), based on Porter's algorithm [10] and one of the most popular stemmers in IR research.

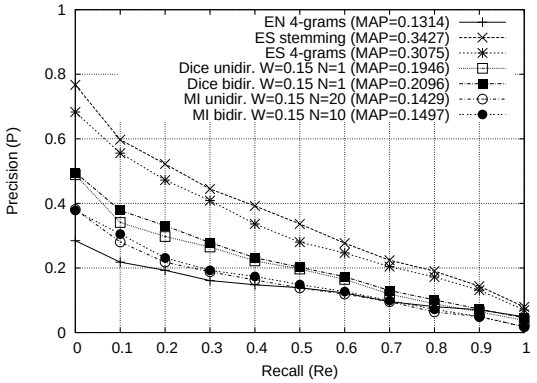

Fig. 3. Final summary Precision vs. Recall graph.

topics split into 4-grams (`ES 4-grams`) —our ideal performance goal. As can be seen, the Dice coefficient in combination with the top-rank-based selection algorithm obtained the best results, performing significantly better than mutual information.

Although we still need to improve our results in order to reach our ideal performance goal, our current results are encouraging, since it must be taken into account that these are our very first experiments, so the margin for improvement is still great.

4 Conclusions and Future Work

This paper describes a system for character n-gram-level alignment in a parallel corpus and its use for direct translation of character n-grams in Cross-Language Information Retrieval. The algorithm proposed consists of two phases. In the first phase, the slowest one, the input parallel corpus is statistically aligned at word-level. In the second phase, n-gram association measures are computed —currently, the Dice coefficient and mutual information—, taking as input the translation probabilities calculated in the previous phase. This solution speeds up the training process, concentrating most of the complexity in the word-level alignment phase, making the testing of new association measures for n-gram alignment easier. On the other hand, two algorithms for the selection of candidate translations have been tested: a top-rank-based algorithm, which takes the N highest ranked n-gram alignments; and a threshold-based algorithm which takes those alignments according to a minimal threshold T.

Our experiments have shown that the Dice coefficient outperforms mutual information. In the case of using the Dice coefficient, the top-rank-based selection algorithm performs better. However, in the case of using mutual information, there is no apparent difference between the two selection algorithms available.

The use of a bidirectional alignment during the input word-level alignment and the introduction of a minimal word-level translation probability threshold have allowed us to reduce drastically both the number of input word alignments to be processed and the number of output n-gram alignments, but without damaging the performance of the system. This way, we can reduce considerably the computing and storage resources required, including processing time. Moreover, these experiments have demonstrated the robustness of the system against noisy or ambiguous input alignments.

With respect to our future work, new tests with other languages of different characteristics are being prepared in order to complete the tune of the system. We will also focus our effort on the development of new algorithms for the selection of candidate translations, and the application of new association measures.

Acknowledgment

This research has been partially funded by Ministerio de Educación y Ciencia and FEDER (TIN2004-07246-C03), Xunta de Galicia (PGIDIT05PXIC30501PN, PGIDIT05SIN044E, *Rede Galega de Procesamento da Linguaxe e Recuperación de Información*, and *Programa de Recursos Humanos* grants), and Universidade da Coruña. The authors desire to thank Prof. John Tait (Univ. of Sunderland, UK) for his support.

References

1. `http://ir.dcs.gla.ac.uk/terrier/`
2. G. Amati and C.J. van Rijsbergen. Probabilistic models of Information Retrieval based on measuring divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.
3. P. Koehn. EUROPARL: A Parallel Corpus for Statistical Machine Translation. In *Proc. of the 10th Machine Translation Summit*, pp. 79–86, 2005. Corpus available in `http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl/`.
4. P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of the 2003 Conf. of the North American Chapter of the ACL*, pp. 48–54, 2003.
5. C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.
6. P. McNamee and J. Mayfield. Character N-gram Tokenization for European Language Text Retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.
7. P. McNamee and J. Mayfield. JHU/APL experiments in tokenization and non-word translation. In vol. 3237 of *LNCS*, pp. 85–97. Springer-Verlag, 2004.
8. A. Nardi, C. Peters, and J.L. Vicedo (eds.). *Working Notes of the CLEF 2006 Workshop*, 2006. Available at `http://www.clef-campaign.org`.
9. F.J. Och and H. Ney. A systematic comparison of various statistical alignment models, 2003. Source code available at `http://www.fjoch.com/GIZA++.html`.
10. Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
11. J. Savoy. Cross-Language Information Retrieval: experiments based on CLEF 2000 corpora. *Information Processing and Management*, 39:75–115, 2003.
12. J. Vilares, M.P. Oakes, and J.I. Tait. CoLesIR at CLEF 2006: rapid prototyping of a N-gram-based CLIR system. In Nardi et al. [8].