Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 95-102. Association for Computational Linguistics.

From Words to Corpora: Recognizing Translation

Noah A. Smith
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218 USA
nasmith@cs.jhu.edu

Abstract

This paper presents a technique for discovering translationally equivalent texts. It is comprised of the application of a matching algorithm at two different levels of analysis and a well-founded similarity score. This approach can be applied to any multilingual corpus using any kind of translation lexicon; it is therefore adaptable to varying levels of multilingual resource availability. Experimental results are shown on two tasks: a search for matching thirty-word segments in a corpus where some segments are mutual translations, and classification of candidate pairs of web pages that may or may not be translations of each other. The latter results compare competitively with previous, document-structure-based approaches to the same problem.

1 Introduction

As in most areas of natural language processing, recent approaches to machine translation have turned increasingly to statistical modeling of the phenomenon (translation models) (Berger et al., 1994). Such models are learned automatically from data, typically parallel corpora: texts in two or more languages that are mutual translations. As computational resources have become more powerful and less expensive, the task of training translation models has become feasible (Al-Onaizan et al., 1999), as has the task of translating (or "decoding") text using such models (Germann et al., 2001). However, the success of the statistical approach to translation (and also to other multilingual applications that utilize parallel text) hangs crucially on the quality, quantity, and diversity of data used in parameter estimation.

If translation is a generative process, then one might consider its reverse process of recognition: Given two documents, might it be determined fully automatically whether they are translations of each other?

The ability to detect translations of a document has numerous applications. The most obvious is as a means to build a parallel corpus from a set of multilingual documents that contains some translation pairs. Examples include mining the World-Wide Web for parallel text (Resnik, 1999; Nie et al., 1999; Ma and Liberman, 1999) and building parallel corpora from comparable corpora such as multilingual collections of news reports. Another use of translation detection might be as an aid in alignment tasks at any level. For example, consider the task of aligning NP chunks (and perhaps also the extra-NP material) in an NP-bracketed parallel corpus; a chunk-level similarity score (Fluhr et al., 2000) built from a word-level model could be incorporated into a framework that involves bootstrapping more complex models of translation from simpler ones (Berger et al., 1994). Finally, reliable cross-lingual duplicate detection might improve performance in n-best multilingual information retrieval systems; at the same time, by detecting the existence of a translation in a multilingual corpus, the cost of translating a document of interest is eliminated.

I present here an algorithm for classifying document pairs as either translationally equivalent or not, which can be built upon any kind of word-to-word translation lexicon (automatically learned or hand-crafted). I propose a score of translational similarity, then describe an evaluation task involving a constrained search for texts (of arbitrary size) that are translation pairs, in a noisy space, and present precision/recall results. Finally, I show that this algorithm performs competitively with the approach of Resnik (1999), in which only

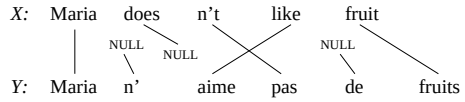


Figure 1: An example of two texts with links shown.

structural information (HTML-markup) is used to detect translation pairs, though the new algorithm does not require structural information.

2 Quantifying Similarity

This section shows how to compute a cross-lingual similarity score, *tsim*, for two texts.[1] Suppose parallel texts are generated according to Melamed's (2000) symmetric word-to-word model (Model A). Let a *link* be a pair (x, y) where x is a word in language $\mathcal{L}_1$ and y is a word in $\mathcal{L}_2$. Within a link, one of the words may be NULL, but not both. The model consists of a bilingual dictionary that gives a probability distribution over all possible link types. In the generative process, a sequence of independent link tokens is generated according to that distribution.

The links are not observed; only the lexical (non-NULL) words in each language are observed. The texts whose similarity score is to be computed, X and Y, correspond to the monolingual lexical projections of the links. For the purposes of this discussion, the texts are viewed as unordered bags of words; scrambling of the link tokens in the two texts is not modeled. An example is illustrated in Figure 1; there are seven link tokens shown, five of which are lexical in X (the English side) and six of which are lexical in Y (the French side).

The next step is to compute the probability of the most probable sequence that could have accounted for the two texts. All permutations of a given link sequence will have the same probability (since the links are generated independently), so the order of the sequence is not important. As noted by Melamed (2000), under the assumption that the quality of a link collection is the sum of the quality of the links, then this problem of finding the best set of links is equivalent to the maximum-weighted bipartite matching (MWBM) problem: Given a weighted bipartite graph $G = (V_1 \cup V_2, E)$ with $|V_1| = |V_2|$ and edge weights $c_{i,j} (i \in V_1, j \in V_2)$, find a matching $M \subseteq E$ such that each vertex has at most one edge in M, and $\sum_{e \in M} c_{i,j}$ is maximized. The fastest known MWBM algorithm runs in $O(ve + v^2 \log v)$ time (Ahuja et al., 1993). Applied to this problem, that is $O(\max(|X|, |Y|)^3)$.

The similarity score should be high when many of the link tokens in the best link collection do *not* involve NULL tokens. Further, it should normalize for text length. Specifically, the score I use is:

$$tsim = \frac{\log \Pr(\text{two-word links in best matching})}{\log \Pr(\text{all links in best matching})} \quad (1)$$

This score is an example of Lin's (1998) mathematical definition of similarity, which is motivated by information theory:

$$sim(X, Y) = \frac{\log \Pr(\text{common}(X, Y))}{\log \Pr(\text{description}(X, Y))} \quad (2)$$

where X and Y are any objects generated by a probabilistic model.[2]

In this research, I seek to show how multiple linguistic resources can be exploited together to recognize translation. The measure in (1) is simplified by assuming that all links in a given translation lexicon are equiprobable. (In some cases I use an automatically induced translation lexicon that assigns probabilities to the entries, but for generality the probabilities are ignored.) This reduces the formula in (1) to

$$tsim = \frac{\#\text{ two-word links in best matching}}{\#\text{ links in best matching}}. \quad (3)$$

Further, to compute *tsim* under the equiprobability assumption, we need not compute the MWBM, but only find the maximum cardinality bipartite matching (MCBM), since all potential links have the same weight. An

[1] I use the term "text" to refer to a piece of text of any length.

[2] Another approach, due to Jason Eisner (personal communication) would be to use a log-likelihood ratio of two hypotheses: joint vs. separate generation of the two texts ($\log \frac{\Pr(\text{all links in the best sequence})}{\Pr(\text{all words in } X)\Pr(\text{all words in } Y)}$). In order to make this value (which is the Viterbi approximation to point-wise mutual information between the two texts) a score suitable for comparison between different pairs of texts, it must be normalized by length. With normalization, this score is monotonic in Lin's (1998) *sim* if a uniform unigram model is assumed for the tokens in the single-language models (the denominator terms).

$O(e\sqrt{v})$ (or $O(|X| \cdot |Y| \cdot \sqrt{|X| + |Y|})$ for this purpose) algorithm exists for MCBM (Ahuja et al., 1993). If the matching shown in Figure 1 is the MCBM (for some translation lexicon), then $tsim(X, Y) = \frac{4}{7}$ under the simplifying assumption.

If Equation (3) is applied to pairs of documents in the *same* language, with a "translation lexicon" defined by the identity relation, then *tsim* is a variant of *resemblance* (r), as defined by Broder et al. (1997) for the problem of monolingual duplicate detection:

$$r(X, Y) = \frac{|S(X) \cap S(Y)|}{|S(X) \cup S(Y)|} \quad (4)$$

where $S(Z)$ is a shingling of the words in Z; a shingling is the set of unique n-gram types in the text for some fixed n (Damashek, 1995). Unlike Broder et al.'s r, however, *tsim* is token-based, incorporating word frequency. Specifically, the intersection of two bags (rather than sets) of tokens contains the minimum count (over the intersected bags) of each type; the union contains the maximum counts, e.g.,

$$\{a, a, a, b, b\} \cap \{a, a, b, b, b\} = \{a, a, b, b\}$$
$$\{a, a, a, b, b\} \cup \{a, a, b, b, b\} = \{a, a, a, b, b, b\}$$

With the assumption of equiprobability, *any* translation lexicon (or, importantly, union thereof) containing a set of word-to-word entries, can be used in computing *tsim*.

3 Finding Translations

Formally, the evaluation task I propose can be described as follows: Extract all translation pairs from a pool of $2n$ texts, where n of them are known to be in language $\mathcal{L}_1$ and the other n are known to be in $\mathcal{L}_2$. Each text can have one or zero translations in the corpus; let the number of true translation pairs be k.

The general technique for completing the task is to first find the best matching of words in text pairs (posed as a bipartite matching problem) in order to compute the *tsim* similarity score. Next, to extract translation pairs of texts from a corpus, find the best matching of texts based on their pairwise *tsim* scores, which can be posed as a "higher-level" MWBM problem: by matching the texts using their pair-wise similarity scores, a corpus of pairs of highly similar texts is extracted from the pool.

If k is known, then the text-matching problem is a generalization of MWBM: Given a weighted bipartite graph $G = (V_1 \cup V_2, E)$ with $|V_1| = |V_2|$ and edge weights $c_{i,j}$, find a matching $M \subseteq E$ *of size* k such that each vertex has at most one edge in M, and $\sum_{e \in M} c_{i,j}$ is maximized. The set of texts in $\mathcal{L}_1$ is V_1, and the set of texts in $\mathcal{L}_2$ is V_2; the weights $c_{i,j}$ are the scores $tsim(v_i, v_j)$. I do not seek a solution to the generalized problem here; one way of approximating it is by taking the top k *tsim*-scored elements from the set M (the MWBM).

If k is not known, it can be estimated (via sampling and human evaluation); I take the approach of varying the estimate of k by applying a threshold τ on the *tsim* scores, then computing precision and recall for those pairs in M whose score is above τ (call this set M_τ):

$$prec_\tau = \frac{|M_\tau \cap T|}{|M_\tau|}, \quad rec_\tau = \frac{|M_\tau \cap T|}{k} \quad (5)$$

where T is the set of k true translation pairs. Performance results are presented as (precision, recall) pairs as τ is lowered.[3]

Melamed (2000) used a greedy approximation to MWBM called competitive linking, which iteratively selects the edge with the highest weight, links those two vertices, then removes them from the graph. (Ties are broken at random.) A heap-based implementation of competitive linking runs in $O(\max(|X|, |Y|) \log \max(|X|, |Y|))$. In the first experiment, I show a performance comparison between MWBM and competitive linking.

4 Experiment: English-Chinese

This experiment used the Hong Kong Hansard English-Chinese parallel corpus. The training corpus is aligned at the sentence level, with segment lengths averaging fifteen words (in each language). The test corpus is aligned at the two-sentence level, with segment lengths averaging thirty words. The first experiment involved ten-fold cross-validation with (for each fold) a training corpus of 9,400 sentence pairs and a test corpus of 1,000 two-sentence pairs. The corpus

[3]The selection of an appropriate τ will depend on the application, the corpus, the lexicons, etc. In my evaluation on WWW data, I use a small development set to choose a threshold that maximizes one measure of performance.

was randomly divided into folds, and no noise was introduced (i.e., $k = n$).[4]

4.1 Translation Lexicon

The main translation lexicon of interest is a union of three word-to-word translation lexicons from different sources. I refer to this translation lexicon as UTL.

The first component translation lexicon, DICT, was made from the union of two English-Chinese electronic dictionaries, specifically, those from Meng et al. (2000) and Levow et al. (2000) (a total of 735,908 entries, many of which are not one-to-one). To make the dictionary exclusively one-to-one entries, each n-to-m entry was processed by removing all function words in either side of the entry (according to a language-specific stoplist), then, if both sides have one or two words (no more), adding all word-pairs in the cross-product (otherwise the entry is discarded).[5] The resulting translation lexicon contains 577,655 word pairs, 48,193 of which contain two words that are present in the corpus. This translation lexicon has the advantage of broad coverage, though it does not generally contain names or domain-specific words, which are likely to be informative, and does not capture morphological variants.

The second translation lexicon, TMTL, is automatically generated by training a symmetric word-to-word translation model (Model A, (Melamed, 2000)) on the training corpus.[6] All word pairs with nonzero probability were added to the translation lexicon (no smoothing or thresholding was applied). On average (over ten folds), this translation lexicon contained 6,282 entries. The TMTL translation lexicons are expected to capture words specific to the domain (Hong Kong government transcripts), as well as common inflections of words, though they will also contain noise.

The third translation lexicon, STR, is the string identity lexicon: (x, y) is in the translation lexicon iff the string x is identical to the string y. This translation lexicon captures punctuation, numerals, alphanumeric strings used to label sections, and English words included as-is in the Chinese corpus. There were 3,083 such pairs of word types in the corpus.

4.2 Filtering

Chen and Nie (2000) note that text pairs that are highly disparate in length are unlikely to be translations. In order to avoid computing *tsim* scores for all pairs in the cross-product, I eliminated all segment pairs whose lengths are outliers in a linear regression model estimated from the training corpus. Earlier experiments (on a different corpus) showed that, if a $(1-p)$-confidence interval is used, the size of the search space reduces exponentially as p increases, while the number of correct translation pairs that do not pass the filter is only linear in p (i.e., the filter gives high recall and high precision). For these experiments, $p = 0.05$; this value was selected based on the results presented in Smith (2001).

4.3 Results

When the length filter was applied to the 1,000,000 possible pairs in the cross-product, 47.9% of the pairs were eliminated, while 94.5% of the correct pairs were kept, on average (over ten folds). *tsim* was computed for each pair that passed the filter, then each matching algorithm (MWBM and competitive linking) was applied. As discussed above, a threshold can then be applied to the matching to select the pairs about whose translational equivalence the score is most confident. Precision and recall plots are shown in Figure 2a. Each line corresponds to a (translation lexicon, matching algorithm) pair, showing average precision and recall over the ten folds as the threshold varies. The plots should be read from left to right, with recall increasing as the threshold is lowered.

When many resources are used, the technique is highly adept at selecting the translation pairs. TMTL alone outperforms DICT alone, probably due to its coverage of domain-specific terms. The competitive linking algorithm lags behind MWBM in most cases, though its performance

[4] It is possible that random division gives a favorable bias in the translation model translation lexicon by increasing the probability that rare words that appear only in certain portions of the corpus will be present in both training and test data.

[5] The limit of two words per side is an arbitrary choice intended to minimize the noise introduced by this processing step.

[6] In parameter estimation, I used the aforementioned MWBM algorithm (instead of Melamed's (2000) competitive linking), which is the maximum posterior approximation to EM. It is not clear, however, that this change yields performance gains.

was slightly better in the case of TMTL. In the case of UTL, for recall up to 0.8251, the thresholded MWBM matching had significantly higher precision than the thresholded competitive linking matching at a comparable level of recall (based on a Sign Test over the ten cross-validation folds, $p < 0.01$).

Table 1 shows the maximum performance (by F-score) for each translation lexicon under MWBM and competitive linking.

4.4 Effects of Noise

Next, I performed an experiment to test the technique's robustness to noise. In this case, the test corpus contained 300 known translation pairs (again, two-sentence texts). From 0 to 2700 additional English texts and the same number of Chinese texts were added. These "noise" texts were from the same corpus and were guaranteed not to be aligned with each other.[7] The length filter eliminated 48.6% of the 9,000,000 possible pairs in the cross-product, keeping 95.7% of the true pairs. The filtered pairs were *tsim*-scored using UTL, then the MWBM was computed. Precision and recall are plotted for various levels of noise in Figure 2b.[8] Only in the highest-noise condition ($\frac{k}{n} = 0.1$) do we observe a situation where a sufficiently strict threshold cannot be used to guarantee an extracted corpus of (nearly) arbitrarily high precision. For example, if 90% precision is required, 88.3%, 60.3%, and 43.7% recall can be guaranteed when $\frac{k}{n}$ is 1, 0.5, and 0.25, respectively.

These experiments show that with a strict threshold this technique is capable of producing a highly precise matching of parallel text from a noisy corpus, though attainable recall levels drop as noise is added. Performance can be boosted by incorporating additional bilingual resources. Finally, even a fast, greedy approximation to the best matching can be useful.

[7] In general, robustness to noise will depend on the source of the noise and how much the noise looks like the true translations. Hence the results presented here may be better or worse than those achieved in specific applications to which this technique might be applied, depending on those factors, filtering, etc.

[8] Experiments were carried out for the TMTL and DICT translation lexicons, and also under competitive linking. Space does not permit a full discussion, though it is worth mentioning that, as in the noiseless experiment, UTL outperformed the others, likewise MWBM outperformed competitive linking.

Tr. lex.	Algorithm	τ	$prec_\tau$	rec_τ	F_τ
UTL	MWBM	0.20000	0.908	0.836	0.871
	CL	0.22078	0.917	0.805	0.857
DICT	MWBM	0.10638	0.776	0.647	0.706
	CL	0.12121	0.770	0.590	0.668
TMTL	MWBM	0.00971	0.841	0.711	0.771
	CL	0.00909	0.854	0.711	0.776

Table 1: Comparison of translation lexicons and matching algorithms at their maximal F-scores. Note that thresholds, and *tsim* scores in general, are comparable only for a given translation lexicon. The STR translation lexicon offered a boost only when used to supplement TMTL ∪ DICT; when added to each alone it had little or no effect.

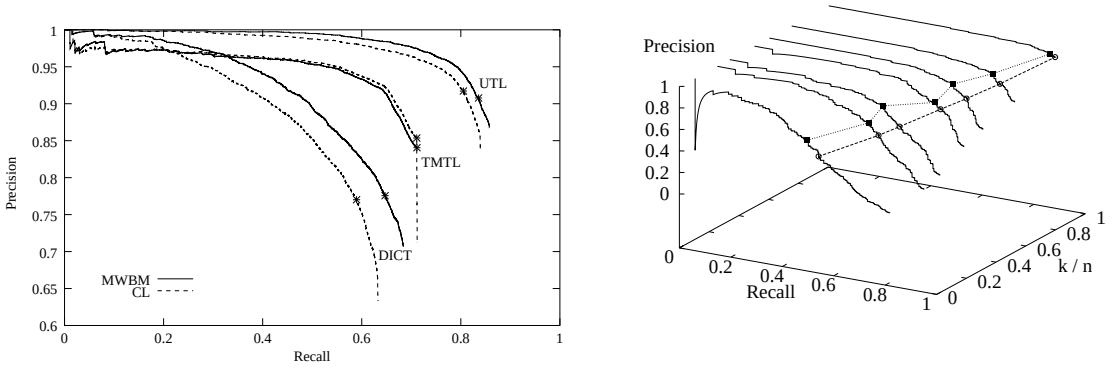
	top 300 pairs		maximum F			
n	*prec*	*rec*	τ	$prec_\tau$	rec_τ	F_τ
300	0.904	0.883	0.16667	0.925	0.863	0.893
400	0.813	0.813	0.25641	0.897	0.787	0.838
500	0.770	0.770	0.28000	0.881	0.717	0.791
600	0.727	0.727	0.28000	0.782	0.707	0.743
900	0.663	0.663	0.32142	0.829	0.600	0.696
1200	0.630	0.630	0.32142	0.733	0.593	0.656
3000	0.483	0.483	0.35849	0.617	0.440	0.514

Table 2: Precision and recall when the top k (300) pairs are taken (i.e., k is known; in the case of $n = 300$, the matching contained only 293 pairs), and at maximal F-scores for various levels of noise.

5 Experiment: English-French

An important application of translation recognition is the construction of parallel text corpora. One source of raw text in this task is the World-Wide Web, for which several parallel text search systems currently exist (Resnik, 1999; Nie et al., 1999; Ma and Liberman, 1999). These systems propose candidate pairs of pages, which are then classified as either translationally equivalent or not. The STRAND system (Resnik, 1999), for example, uses structural markup information from the pages, without looking at their content, to attempt to align them.

If the *tsim* technique can provide a classifier that rivals or complements the structural one, using as it does an entirely orthogonal set of features, then perhaps a combined classifier could provide even greater reliability. In addition, custom-quality parallel corpora could be generated from comparable corpora that lack



(a) (b)

Figure 2: (a.) **Precision and recall with no noise.** This plot shows precision and recall averaged over all ten folds. Each point corresponds to a threshold value; the threshold becomes less strict from left to right. Shown are curves for each of UTL, TMTL, and DICT under both algorithms (MWBM, CL); the maximum F scores are marked (see Table 1). (b.) **Precision-recall curves at varying levels of noise.** $k = 300$ in all cases; the circles and dashed line show precision and recall for the top 300 pairs in the matching (i.e., if k were known, it would not make sense to use a lower threshold, so the only reasonable thresholds are to the left), and the squares and dotted line show precision and recall at each condition's maximum F-score—the values are shown in Table 2. (Note that the curves "stop" before reaching a point where recall is 1.0, since a point is eventually reached where no more matches are possible (because of filtering).)

structural features. This experiment also shows that *tsim* is scalable to larger texts.

5.1 Translation Lexicon

In this experiment, the language pair is English-French. Multiple sources for the translation lexicon are used in a manner similar to Section 4.1.

- An English-French dictionary (a total of 34,808 entries, 4,021 of which are not one-to-one).[9] It contains morphological variants but does not include character accents. Each n-to-m entry was processed by stoplisting and then extracting all word-pairs in the remaining cross-product as in section 4.1. Result: 39,348 word pairs, 9,045 of which contain two words present in the corpora.
- A word-to-word translation model (Melamed, 2000) trained on a verse-aligned Bible using MWBM (15,548 verses, averaging 25.5 English words, 23.4 French words after tokenization). Result: 13,762 word pairs.
- English-French cognate pairs, identified using the method of Tiedemann (1999). Space does not permit a full description of the technique; I simply note that cognates were identified by thresholding on a specially-trained string-similarity score based on language-specific character-to-character weights.[10] Result: 35,513 word pairs. An additional set of 11,264 exact string matches were added. These entries are quite noisy.

The union of these translation lexicons consists of 68,003 unique word pairs. The experiment used only this union translation lexicon.

5.2 Results

In order to compare *tsim* with structural similarity scoring, I applied it to 325 English-French web-document pairs. These were the same pairs for which human evaluations were carried out by Resnik (1999).[11] Note that this is not a matching task; the documents are presented as candidate pairs, and there is no competition among pages for matches in the other language. At different thresholds, a κ score of agreement (with each of Resnik's (1999) two judges and their

[9] This dictionary was generated using a dictionary derived from one available at http://www.freedict.com.

[10] Tiedemann trained these weights using a list of known cognates; I use a noisy list of weighted translation pairs (specifically, TMTL) Hence the resources required to extract cognates in this way are no different from those required for the translation model.

[11] One additional pair was thrown out because it contained compressed data; it is assumed that pair would not pass a language identification filter.

intersection) may be computed for comparison with Resnik's STRAND system, along with recall and precision against a gold standard (for which I use the intersection of the judges—the set of examples where the judges agreed). Note that recall in this experiment is relative to the candidate set proposed by the STRAND search module, not the WWW or even the set of pages encountered in the search.

The estimate of *tsim* (MWBM on the words in the document pair) is not computationally feasible for very large documents and translation lexicons. In preliminary comparisons, I found that representing long documents by as few as their first 500 words results in excellent performance on the κ measure. This allows $O(1)$ estimation of *tsim* for two documents: look only at the first (fixed) n words of each document. Further, the competitive linking algorithm appears to be as reliable as MWBM. The results reported here approximated *tsim* in using competitive linking on the first 500 words.

Of the 325 pairs, 32 were randomly selected as a development set. Maximizing κ on this set yielded a value of $\tau = 0.15$.[12] κ scores against each judge and their intersection were then computed at that threshold on the test set (the remaining 293 pairs). These are compared to κ scores of the STRAND system, on the same test set, in Table 3. In every case, the *tsim* classifier agreed more strongly with the human evaluations.

At $\tau = 0.15$, precision was 0.680 and recall was 0.921, $F = 0.782$ (on the same set, STRAND structural classification achieved 0.963 precision and 0.684 recall, $F = 0.800$). Figure 3 shows κ, precision, and recall plotted against τ.

6 Future Directions

The success of this approach suggests a way to construct parallel corpora from any large, segmented comparable corpus: start with a translation model estimated on a small, high-quality parallel text, and a core dictionary; then extract document pairs with high similarity (*tsim*) and add them to the parallel corpus. Next, estimate word-level translational equivalence empirically from the enlarged corpus and update the translation lexicon; extract documents iteratively. The experiments presented here show that, even in highly noisy search spaces, *tsim* can be used with a threshold to extract a high-precision parallel corpus at moderate recall.

[12] One could select such a threshold to maximize any objective function over the development set.

Comparison	N	Pr(Agree)	κ
J1, J2	245	0.98	0.96
J1, STRAND	250	0.88	0.70
J2, STRAND	284	0.88	0.69
J1 ∩ J2, STRAND	241	0.90	0.75
J1, $tsim(\tau = 0.15)$	249	0.92	0.83
J2, $tsim(\tau = 0.15)$	283	0.92	0.82
J1 ∩ J2, $tsim(\tau = 0.15)$	240	0.93	0.85

Table 3: Comparison with STRAND. The test set is 294 of the 326 pairs in Resnik's (1999) test set. The STRAND κ scores are similar to those published by Resnik (1999). The 32 development pairs were used to select the 0.15 threshold. N is the number of examples for which judgement-comparison was possible in each case (human judges were sometimes undecided; those cases are ignored in computing κ).

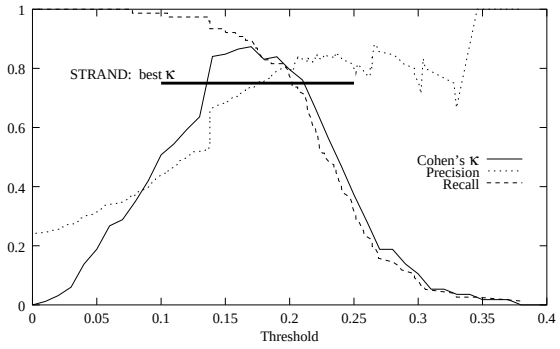


Figure 3: Performance measures as functions of the threshold τ: the κ agreement score with the two judges' intersection, precision, and recall. All measures are on the test set. The κ score obtained by STRAND is shown as well.

It is worth noting that the STRAND classifier and the *tsim* classifier disagreed 15% of the time on the test set. A simple combination by disjunction (i.e., "(X, Y) is a translation pair if either classifier says so") yields precision 0.768, recall 0.961, $F = 0.854$, and κ (with the judges' intersection) at 0.878. In future work, more sophisticated combinations of the two classifiers might integrate the advantages of both.

7 Conclusion

I have proposed a language-independent approach to the detection of translational equivalence in texts of any size that works at various bilingual resource levels. Fast, effective approximations have also been described, suggesting scalability to very large corpora. Notably, *tsim* is adaptable to any probabilistic model of translational equivalence, because it is an instance of a model-independent definition of similarity. The core of the technique is the computation of optimal matchings at two levels: between words, to generate the *tsim* score, and between texts, to detect translation pairs.

I have demonstrated the performance of this technique on English-Chinese and English-French.[13] It is capable of pulling parallel texts out of a large multilingual collection, and it rivals the performance of structure-based approaches to pair classification (Resnik, 1999), having better κ agreement with human judges.

Acknowledgements

This work was supported in part by the National Science Foundation and DARPA/ITO Cooperative Agreement N660010028910 (at the University of Maryland) and a Fannie and John Hertz Foundation Fellowship. The author thanks Dan Melamed, Philip Resnik, Doug Oard, Rebecca Hwa, Jason Eisner, Hans Florian, and Gideon Mann for advice and insightful conversations; also Gina Levow for making available the bilingual dictionaries and Philip Resnik for sharing the STRAND test data and human judgements.

References

R. K. Ahuja, T. L. Magnati, and J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ.

Y. Al-Onaizan, J. Cuřin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, N. A. Smith, F.-J. Och, D. Purdy, and D. Yarowsky. 1999. Statistical Machine Translation. Technical report, Johns Hopkins University.

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, J. D. Lafferty, R. L. Mercer, H. Printz, and L. Ureš. 1994. The Candide system for machine translation. In *ARPA Workshop on Speech and Natural Language Technology*, pages 157–163. Morgan Kaufman.

A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. 1997. Syntactic clustering of the Web. In *Sixth International World-Wide Web Conference*, Santa Clara, CA.

J. Chen and J.-Y. Nie. 2000. Web parallel text mining for chinese-english cross-language information retrieval. In *International Conference on Chinese Language Computing*, Chicago, IL.

M. Damashek. 1995. Gauging similarity with n-grams: language independent categorization of text. *Science*, 267:843–8.

C. Fluhr, F. Bisson, and F. Elkateb. 2000. Mutual benefit of sentence/word alignment and cross-lingual information retrieval. In Véronis, J. (ed.), *Parallel Text Processing*. Kluwer Academic Publishers, Dordrecht.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *39th ACL*, Toulouse, France.

G.-A. Levow, D. W. Oard, and C. I. Cabezas. 2000. Translingual topic tracking with PRISE. In *Topic Detection and Tracking Workshop*, Tysons Corner, VA.

D. Lin. 1998. An information-theoretic definition of similarity. In *International Conference on Machine Learning*, Madison, WI.

X. Ma and M. Liberman. 1999. BITS: a method for bilingual text search over the web. In *Machine Translation Summit VII*, Singapore.

I. D. Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

H. Meng, B. Chen, E. Grams, S. Khudanpur, G.-A. Levow, W.-K. Lo, D. W. Oard, P. Schone, H.-M. Wang, and J. Wang. 2000. Mandarin-English: Investigating Translingual Speech Retrieval. Technical report, Johns Hopkins University.

J. Nie, P. Isabelle, M. Simard, and R. Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *ACM-SIGIR Conference*, pages 74–81, Berkeley, CA.

P. Resnik. 1999. Mining the Web for bilingual text. In *37th ACL*, College Park, MD.

N. A. Smith. 2001. Detection of Translational Equivalence. Undergraduate honors thesis, University of Maryland.

J. Tiedemann. 1999. Automatic construction of weighted string similarity measures. In *Conference on EMNLP and VLC*, College Park, MD.

[13] Comparable experiments using another version of the score showed performance for English-Spanish on the matching task to be even better than for English-Chinese (using that same score) (Smith, 2001).