Automatic Identification of Parallel Documents with light or without Linguistic Resources

Alexandre Patry and Philippe Langlais

Laboratoire de Recherche Appliquée en Linguistique Informatique
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
http://rali.iro.umontreal.ca

Abstract. Parallel corpora are playing a crucial role in multilingual natural language processing. Unfortunately, the availability of such a resource is the bottleneck in most applications of interest. Mining the web for parallel corpora is a viable solution that comes at a price: it is not always easy to identify parallel documents among the crawled material. In this study we address the problem of automatically identifying the pairs of texts that are translation of each other in a set of documents. We show that it is possible to automatically build particularly efficient content-based methods that make use of very little lexical knowledge. We also evaluate our approach toward a front-end translation task and demonstrate that our parallel text classifier yields better performances than another approach based on a rich lexicon.

1 Introduction

Parallel corpora are currently playing a crucial role in multilingual natural language processing applications. Aligned at the sentence level, a task that has been shown to be fairly easy [1], a parallel corpus turns out to be already very useful for bilingual concordancers [2] and is the cornerstone of most of the commercial translation memory systems that have been and still are popular among professional translators.

Aligned at the word level, a task for which we have practical and now well understood techniques [3], a parallel corpus may be useful for many applications such as machine translation, word-sense disambiguation and cross-lingual information retrieval.

Few reasonably large, well organized *bitexts* (bilingual corpora where translation relations are explicitly marked) are in common use within the NLP community. The canonical example of such a resource is the so-called *Hansard*, that is, the Canadian parliament debates in both French and English. More and more bitexts of different quality and size are also available for various pairs of languages, among them the Chinese-English Hongkong Hansard, the proceedings of the European Parliament in twenty languages[1], as well as an English-Inuktitut

[1] http://www.europarl.eu.int/home/default_fr.htm

Hansard[2] [4]. Other resources, such as the Bible, are translated in many different languages (but not necessarily organized into bitexts), and have shown some practical usefulness in recent machine translation tasks [5].

However, it is widely acknowledged that the availability of parallel corpora is the bottleneck in many applications of interest. The known available parallel corpora are of limited domain (*viz* legislative and newswire texts) and are mostly available for few well-represented language pairs. Several approaches have been proposed to overcome bitext sparseness, among them mining the web for parallel data [6–8], making use of *comparable corpora* (texts that are related without necessarily being translations) [9], as well as the challenging issue of using totally unrelated corpora in two different languages [10].

The present study is intended to be applied as a post-processing stage after a set of documents has been collected (for instance from the Internet). In section 2, we address the problem of automatically identifying parallel documents in a (likely noisy) set of texts. Doing so, we explore different approaches to the task that make use of few or no specific linguistic resources. In sections 3 and 4 we evaluate our approaches on two tasks: a controlled one on a part of the EUROPARL corpus, as well as a real task we faced when developing an English-Spanish concordancer. We then show that some of the approaches we investigated are very effective at identifying parallel texts and that their use for seeding a translation engine is also fruitful.

2 Methodology

Acquiring a bitext from the web requires several steps that have been carefully described in [6], the first of which consists in crawling Internet sites in order to download more or less any document that could be converted into a plain text file. We consider this step already done. We further assume that what comes out from the web crawling process is two sets of documents (a source set $\mathcal{S}$ and a target set $\mathcal{T}$). The identification of the language of a document might be carried out automatically if not available.

For now on, we will assume that a text is simply an element of a set with no specific external information attached to it such as its name or its url. This precludes the use of name-based heuristics to pairing up the texts, such as the ones described in [7]. Our motivation for this does not lie in an aesthetic way of thinking, but corresponds to an attempt to evaluate as objectively as possible different linguistically poor content-based metrics. In any case, name-based filtering could be introduced as a preprocessing stage or could as well be considered a feature of the classifier we describe in section 2.2.

This being said, the identification of pairs of parallel texts is accomplished in two steps: the scoring of all the pairs of the Cartesian product $\mathcal{S} \times \mathcal{T}$ and the labelling of each pair as a parallel or not, on the basis of those scores. We now describe in section 2.1 the different content-based metrics we considered, and in section 2.2 the decision process we devised.

[2] `http://www.inuktitutcomputing.ca/NunavutHansards/`

2.1 Content-Based Metrics

We considered three types of metrics to measure the similarity of two documents.

Cosine Measure The cosine measure (COSINE) is a classical one in information retrieval and quantifies the similarity of two vectors. It is expressed by:

$$cos(v_1, v_2) = \frac{v_1 \cdot v_2}{||v_1||\,||v_2||} \tag{1}$$

where v_1 and v_2 are the two vectors to be compared. It takes values between zero and unity, where a greater value means a greater similarity. We followed the approach of Nadeau and Foster [11] and represented a document by a vector whose dimension expresses the number of different tokens in the corpus. One vector is built for each of the following feature families:

- Numbers (NUMBER): any sequence of digits.
- Selected punctuations (PUNCT): parenthesis, square brackets and double quote.
- Named entity (NAME): any capitalized word that is not the first in a sentence.

It has to be noted that this set of tokens is fairly language independent and would a priori apply well for many pairs of languages. A possible extension would be to add a feature family that contains all words that are entries in a bilingual lexicon.

Normalized Edit Distance As mentioned by Nadeau and Foster [11], a bag of words representation is a rough approximation of a document. In order to improve on this hypothesis, they suggested to use the so-called *edit-distance* (EDIT) [12]. Each document is now treated as a sequence of features and the edit-distance between two sequences (that is, the minimal number of insertions, deletions and substitutions required to transform the first sequence into the second one) quantifies the similarity of the associated documents; the smaller the edit-distance is, the greater is the similarity of the documents. In order to work around the fact that the edit distance depends on the sequence length, we normalize it by the length of the longest of the two sequences.

We compute the normalized edit-distance on the same feature families as those used with the cosine measure. We show in Figure 1 an example of a feature vector and a sequence of named entities on a quotation from the EUROPARL corpus.

Alignment Scores Another natural candidate to evaluate the parallelness of two documents is the output of a sentence aligner. A sentence aligner takes a pair of parallel documents as inputs and tries to pair sentences that are translations of each others. We used for that purpose the JAPA aligner which performs well and fast [1]. By default, JAPA produces a sequence of alignments whose patterns

In conclusion, while key infrastructure projects have been supported by the *European*$_3$ *Regional*$_5$ *Development*$_2$ *Fund*$_4$ and the *Cohesion*$_1$ *Fund*$_4$, we should remember that the *European*$_3$ *Social*$_6$ *Fund*$_4$ has played a very important role in helping the less well-off in our society.

Named entities feature vector	$(1, 1, 2, 3, 1, 1)$
Named entities sequence	"European", "Regional", "Development", "Fund", "Cohesion", "Fund", "European", "Social", "Fund"

Fig. 1. A feature vector and a sequence of named entities as used to compute the cosine measure and the edit-distance. The named entities are italicized and indexed with their position in the feature vector.

belong to the set *0-1*, *1-0*, *1-1*, *1-2*, *2-1* and *2-2* (a *1-2* pattern indicates that one source sentence is aligned with two target ones). Two documents that are parallel should present many one-to-one (*1-1*) patterns, while documents that are not should contain many insertion (*0-1*) or deletion (*1-0*) patterns. In addition, JAPA produces an alignment cost (ACOST) which measures the overall quality of the alignment.

Five scores are computed with the aligner output : the ratio of *0-1* and *1-0* alignments, the ratio of *1-1* alignments, the ratio of *1-2* and *2-1* alignments, the ratio of *2-2* alignments and ACOST. We named the group of the four ratios M-N.

2.2 Decision Process

Once a set of scores is associated with a pair of documents, all we need to do is decide whether or not they are translations of each other. We could set up a threshold based approach, but instead we trained an AdaBoost [13] classifier. The training process takes as input scored pairs of texts labelled as parallel or not. It then builds a function that will take a scored pair as input and output whether it is parallel or not.

AdaBoost is a learning algorithm that combines many weak classifiers[3] into a stronger one. It achieves this by training weak classifiers successively, each time focusing on examples that have been hard to classify correctly by the previous weak classifiers. In our experiments, we bounded the number of weak classifiers to 75 and used neural networks [14] with one hidden layer of five units as weak classifiers. Training and testing was done with the PLEARN software[4].

Because the classifier is trained on all the pairs of the Cartesian product $\mathcal{S} \times \mathcal{T}$, the ratio of parallel pairs is very low. To circumvent this imbalance, the examples are weighted to assign 50% of the probability mass to the parallel pairs. The training algorithm is described in Figure 2.

[3] The only constraint on a weak classifier is that it must be right more than of the time.

[4] More informations on PLEARN can be found at `http://plearn.sourceforge.net`

Inputs : $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ the training set of scored and labelled document pairs where $x_i \in \mathcal{R}^d$ is a vector containing the d observed scores for the i^{th} document pair and $y_i \in \{\mathit{parallel}, \mathit{not\text{-}parallel}\}$ is the label of the i^{th} pair.

1. Initialize the weight of each document pair for the training process such that non-parallel and parallel pairs have the same total weight

$$P_1(x_i, y_i) \leftarrow 0.5 \cdot \frac{1}{|\{(x, y) \in \mathcal{D} : y = y_i\}|}$$

2. For each round $t \leftarrow 1 \ldots T$
   (a) Train a small neural network $h_t : \mathcal{R}^d \rightarrow \{\mathit{parallel}, \mathit{not\text{-}parallel}\}$ that will take as input a scored document pair and classify it as parallel or not. The small neural network is trained on the data $\mathcal{D}$ and their weight P_t.
   (b) Compute the weighted ratio of the document pairs misclassified by $h_t(\cdot)$

$$\epsilon_t = \sum_{\{(x_i, y_i) \in \mathcal{D} : y_i \neq h_t(x_i)\}} P_t(x_i, y_i)$$

   (c) If $\epsilon_t \geq 0.5$ then $T \leftarrow t$ and goto 3
   (d) Compute the weight of the vote of $h_t(\cdot)$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$$

   (e) Compute P_{t+1}, the weight of each document pairs for the next iteration, emphasizing on examples that have been misclassified by $h_t(\cdot)$

$$P_{t+1}(x_i, y_i) = \frac{P_t(x_i, y_i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

   where Z_t is a normalisation factor chosen such that $\sum_{(x_i, y_i) \in \mathcal{D}} P_{t+1}(x_i, y_i) = 1$
3. Return the strong classifier which performs a weighted vote on the small neural networks $(h_1(\cdot), h_2(\cdot), \ldots, h_T(\cdot))$

$$H(x) = \operatorname*{argmax}_{type} \sum_{\{t \in [1,T] : h_t(x) = type\}} \alpha_t$$

   where x is a scored document pair and $\mathit{type} \in \{\mathit{parallel}, \mathit{not\text{-}parallel}\}$.

Fig. 2. The AdaBoost algorithm used to train the parallel pair classifier. In our experiments, T was set to 75 and the $h_t(\cdot)$ were neural nets with one hidden layer of five units.

3 Controlled Task

3.1 Corpus

EUROPARL is a large corpus of bitexts drawn from the European Parliament between April 1996 and September 2003 [15]. It includes versions of the documents in 11 languages, but we focus in this study on the English-Spanish bitext. Our test corpus contains 487 English documents (therefore 487 Spanish ones), thus summing to 237,169 potential pairs of documents. Each document contains an average of about 2,800 sentences.

3.2 Evaluation Protocol

The task was to identify the parallel documents in our corpus using the scores we discussed earlier. Since we had to train a classifier, we applied five fold cross-validation. The set of examples was partitioned into five subsets and five experiments were run, each time testing with a different subset and training with the remaining examples.

Since the EUROPARL corpus is already aligned at the document level, it is straightforward to determine *precision* and *recall*, as well as the *f-measure* (harmonic mean of both). Precision (resp. recall) is the ratio of pairs of documents correctly identified as parallel over the total number of pairs identified as such (resp. over the total number of parallel documents in the corpus).

3.3 Reference system

In order to assess the performance of the different classifiers we trained, we implemented a fair reference system which makes use of a *bilingual lexicon* (a set of pair of words that are translations of each other). This variant is named DICTIONARY hereafter. We downloaded the Spanish-English dictionary of the PYTHOÑOL project[5], a project devoted to helping English speakers learn Spanish. This dictionary contains more than 70 000 bilingual entries.

We represent a document by the set of its words that are less frequent than a given threshold (the value was set to 2 in this study). We explore the Cartesian product $\mathcal{S} \times \mathcal{T}$ following a greedy strategy. For each source document $s = \{s_i\}_{i \in [1,N]}$, we sort the target ones $t = \{t_j\}_{j \in [1,M]}$ according to the number of glosses found in the dictionary D between the words representing s and t (equation 2) and pick the best-ranked target document. Note that in the eventuality of two source documents paired to the same target one, we simply remove the two pairs from the bitext[6].

$$\frac{1}{N+M} \times \sum_{i \in [1,N]} \sum_{j \in [1,M]} \delta((s_i, t_j) \in D) \tag{2}$$

[5] `http://sourceforge.net/projects/pythonol/`

[6] This did not happen in the EUROPARL experiment.

Table 1. Average precision, recall and f-measure (f_1) figures as a function of the scores the classifier has been trained on. Note that because these figures are averaged over a five folds cross-validation, the f_1 is not always coherent with the reported precision and recall. Ratios are expressed as percentages for more readability.

Configuration							Performances		
COSINE	EDIT	NUMBER	PUNCT	NAME	ACOST	M-N	precision	recall	f_1
	√	√	√	√			100	100	100
√	√	√	√	√	√	√	99.8	99.8	99.8
	√	√					98.3	99.8	99.0
	√	√			√		96.6	99.8	98.1
					√		85.8	99.8	92.1
						√	65.6	99.4	77.1
					√	√	49.3	99.4	62.7
√		√	√	√			24.6	99.2	38.7
√		√					12.4	98.9	21.8

3.4 Results

We trained classifiers on various combinations of the scores we described in section 2.1. The performances of each classifier are reported in Table 1. The reference system did as well as our best configuration : a perfect score. The different configurations all performed almost equally on recall, but can be distinguished on their precision figure. The better performance of the classifiers which use the normalized edit-distance instead of the cosine measure leads us to believe that feature ordering is important when searching for parallel documents in a corpus like EUROPARL (long documents carefully translated).

The configurations using the alignment ratios did not perform well. They had an average f-measure 20% lower than the best metrics and were unstable across the five fold cross-validation process. While the f-measures of the four best configurations had a standard deviation lower than 3%, the configurations using alignment types had a standard deviation ranging from 15% to 18%.

We also observe that the normalized edit-distance on numbers alone (third line of Table 1) achieve a f-measure of 99%. This suggests that numbers are very good indicators of parallelism for this kind of corpus. Indeed, parliamentary proceedings contains many stable numbers like dates, law numbers and counts of votes. So our approach could be use with languages where named entities are not trivial to extract.

Last but not least, it is interesting to note that the best classifier we devised performs perfectly on this task, as did the dictionary variant, but without requiring any specific bilingual lexicon.

4 Real World Task

In response to frequent requests, we decided at RALI to extend TSRALI.COM[2], our bilingual concordancers, to the Spanish-English language pair. At that time, the RALI had an agreement with the Pan American Health Organization (PAHO) to create a *transbase* (a bitext searchable online via TSRALI.COM) out of the texts on their web site[7]. A priori, mining a web site in order to extract parallel texts is fairly easy, but in fact, it turned out to be a tricky task [16]. There was no clear hierarchy to rely on for pairing up the documents and the naming conventions were too inconsistent for identifying the language of each text: in short, a perfect test for our system!

4.1 Corpus

SILC[8] was run to discover the language of each file downloaded from the PAHO website, leaving us with 2,523 files identified as English and 4,355 ones as Spanish. Each document contains an average of about 180 sentences. Casual inspection of this material reveals that many files were duplicates (or close duplicates) and that some texts were bilingual. This is however the material we considered, which means that our classifier had to select parallel pairs among over 10 million candidates.

For testing purposes, we downloaded a bitext from the PAHO web site that was written one year after we collected the corpus mentioned above. This document was aligned at the sentence level by JAPA. Following a usual procedure, non *1-1* alignments were removed and the remaining pairs were manually checked for parallelness. A total of 520 pairs of sentences was thus obtained.

4.2 Evaluation Protocol

We devised an evaluation protocol different from the one we discussed in section 3.2. We now want to measure the usefulness of our approach for a real task, namely statistical machine translation (SMT). The reasons for this choice are twofold. First, the identification of parallel documents only makes sense when applied to a front-end (bilingual) application, and machine translation is the bilingual application par excellence. Furthermore, the building of a statistical translation engine is entirely automatic once a bitext is identified. The second reason for evaluating a front-end task lies in the fact that we do not have a clear gold standard bitext against which to evaluate our approach.

We applied the following protocol. We trained our classifiers on the EUROPARL corpus and used each of them to identify parallel document pairs in PAHO. Each set of parallel document pairs was then filtered to remove pairs sharing a document with another pair. This filtering step was introduced to remove uncertain pairs. Each filtered set of document pairs was then used to train a Spanish to English translation engine with which we translated the 520 test sentences (see section 4.1). The quality of the different configurations was evaluated by comparing the automatically translated Spanish sentences with the one we downloaded by

[7] http://www.paho.org

[8] Information on SILC can be found at http://rali.iro.umontreal.ca

applying automatic metrics that are commonly used within the machine translation community (see section 4.4).

In addition to the evaluation using a front end task, we manually checked the precision (see section 3.2) of each configuration.

4.3 SMT

Our SMT engine follows the noisy channel paradigm introduced for machine translation by *Brown et al.* [3]. It can be characterized abstractly as follows:

$$\hat{e} = \operatorname*{argmax}_{e \in \mathcal{E}} p(e|s) = \operatorname*{argmax}_{e \in \mathcal{E}} p(s|e)p(e) \tag{3}$$

where $\hat{e}$ is the (English) translation we seek for a (Spanish) sentence s, and where $p(s|e)$ and $p(e)$ are the translation and language model respectively. The translation model tells us which words should be translation of each other, without necessarily knowing their final position in the translation, while the language model captures some knowledge on the fluency of a sequence of (English) words.

We followed the procedure described in [17] to train both models, and relied on the PHARAOH decoder [18] to perform the argmax operation.

4.4 Metrics

We used different metrics to evaluate the quality of the automatically produced translations. Each metric has its own strengths, the discussion of which is not the purpose of the present exposure. They all compare the candidate translations to a gold standard (in our case a human translation).

The two first metrics are error rates (the lower the better). SER (for Sentence-Error-Rate) is the percentage of sentences produced that are different from the gold standard. WER (for Word-Error-Rate) is the normalized edit-distance between a produced translation and its reference (a rate of 0 would express a perfect translation, a rate of 100, a maximally bad translation).

BLEU and NIST are precision metrics (the higher the better) that, roughly speaking, count the number of sequences that a translation shares with its reference, giving more credit to longer sequences. We used the script `mteval` available at the NIST web site[9] to compute those scores. The BLEU metric ranges between 0 and 1 (1 qualifying the reference itself), while the NIST score is not normalized and the reference itself would be rated 13.11.

4.5 Results

We compared the performance of our translation engine when trained on four different bitexts. The DICTIONARY one was obtained by the reference system described in section 3.3, EDIT is the bitext identified by the best-ranked classifier

[9] `http://www.nist.gov/speech/tests/mt/mt2001/resource`

in the EUROPARL task (line 1 of Table 1), and COSINE is the classifier we trained on the cosine score on the same features (line 8 of Table 1).

Contrary to our former experiment, COSINE and EDIT performed similarly well. This could be explained by the shorter length of the documents and by the filtering step applied on the parallel pairs, which seems to eliminate untrusted pairs. Inspection of their bitexts showed that they shared only 229 document pairs, so we trained another translation engine on the union of those bitexts. The scores of all the translation engines are reported in Table 2. We observe that the performance of the engine trained on the bitexts identified by our COSINE ∪ EDIT classifiers is better than the performance of the engine trained on the DICTIONARY bitext. This is a very satisfactory result since no manual intervention was required, neither any special bilingual resource. Another encouraging result is that the precision of all our classifiers is very high (99% or greater).

Table 2. Evaluation of our parallel text identification procedure through a machine translation task where N is the number of document pairs identified as parallel. See the text for more.

bitext	N	SER	WER	NIST	BLEU	precision
COSINE ∪ EDIT	494	99.42	60.02	5.3125	0.2435	99.0
DICTIONARY	529	99.42	61.67	5.1989	0.2304	89.2
EDIT	390	99.42	61.53	5.1342	0.2290	99.0
COSINE	333	99.23	62.23	5.1629	0.2256	99.7

5 Related Work

This study was inspired by the work of Nadeau and Foster [11] who suggested viewing a document as bags of features such as proper names, numbers, and the like. They have shown that coupled with a fairly tolerant filter on the date of issue of a news story, they could align with high accuracy the Canada Newswire news feed[10]. They also mention that considering word order in a document would be a better idea.

Our work, although independently developed, resembles that of Munteanu and al. [9]. In their study, the authors showed that a translation engine could benefit from parallel sentences automatically extracted from comparable corpora. The approach they propose is analogous to the one we described here, basically training a classifier (in their case via a maximum entropy approach) to identify pairs of sentences (while we look for pairs of documents). To do so, they relied on more extensive resources than what we considered here. For instance, they assumed the availability of a bitext in order to train a translation

[10] `http://www.newswire.ca`

model that they used for aligning sentences at the word level. In fact, both approaches have their own merits and specificities of application. Our approach would be more suited for corpora where we know a priori that many documents are parallel.

6 Future Work and Conclusions

We have extended the approach of Nadeau and Foster [11] in three different ways. First, we tested this idea on different corpora that might not be as friendly as news are. As a matter of fact, news inherently contain a lot of named entities and dates. Second, we experimented with whether maintaining the order of the features in the text would be beneficial. We showed that doing so can surpass or complement the bags of words representation. Third, we tested the impact of such methods on a real task: machine translation. We demonstrated that an approach using poor lexical metrics yields better results than a fair one relying on a rich lexicon.

The approach we propose is highly flexible because it relies on an automatic training procedure which allows us to easily integrate new features to describe a document. In this study, we decided to rely on features such as numbers and named entities; but we observed that the number of hits in a bilingual dictionary may also be a good feature for pairing documents. This could be added to our feature list.

In this study, we systematically considered the Cartesian product of the source and target document sets. This does not come without a computational load. We could also apply a risk-less pre-filtering stage following ad-hoc strategies, such as the length-based criterion proposed by [6].

Finally, we would like to investigate the impact of computing the features on only a part of the documents (for instance the first few sentences). This would speed up the edit-distance computation and therefore the overall process.

Acknowledgements

We would like to thank Marie Ouimet and Leïla Arras who provided us with the PAHO documents we used in this study. Elliott Macklovitch and Ol'ga Feiguina made some useful comments on the first draft of this document. This work has been subsidized by NSERC and FQRNT.

References

1. Langlais, P., Simard, M., Veronis, J.: Methods and practical issues in evaluating alignment techniques. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL), Montréal, Quebec, Canada (1998) 711–717

2. Macklovitch, E., Simard, M., Langlais, P.: Transsearch: A free translation memory on the world wide web. In: Second International Conference On Language Resources and Evaluation (LREC). Volume 3., Athens Greece (2000) 1201–1208
3. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1993) 263–311
4. Martin, J., Johnson, H., Farley, B., Maclachlan, A.: Aligning and using an english-inuktitut parallel corpus. In: HLT-NAACL Workshop: Building and Using Parallel Texts - Data Driven Machine Translation and Beyond, Edmonton, Canada (2003) 115–118
5. Oard, D.W., Och, F.J.: Rapid-reponse machine translation for unexpected languages. In: Machine Translation Summit IX, New Orleans, Louisiana, USA (2003)
6. Kraaij, W., Nie, J.Y., Simard, M.: Embedding web-based statistical translation models in cross-language information retrieval. Computational Linguistics **29** (2003) 381–419
7. Resnik, P., Smith, N.A.: The web as a parallel corpus. Computational Linguistics **29** (2003) 349–380 Special Issue on the Web as a Corpus.
8. Ma, X., Liberman, M.: Bits: A method for bilingual text search over the web. In: Machine Translation Summit VII, Kent Ridge Digital Labs, National University of Singapore (1999)
9. Munteanu, D.S., Fraser, A., Marcu, D.: Improved machine translation performance via parallel sentence extraction from comparable corpora. In: HLT-NAACL. (2004) 265–272
10. Rapp, R.: Automatic identification of word translations from unrelated english and german corpora. In: Proceedings of the 37th conference on Association for Computational Linguistics, Association for Computational Linguistics (1999) 519–526
11. Nadeau, D., Foster, G.: Real-time identification of parallel texts from bilingual news feed. In: CLINE 2004, Computational Linguistics in the North East (2004) 21–36
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl. **6** (1966) 707–710
13. Y.Freund, Schapire, R.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence **14** (1999) 771–780 Appearing in Japanese, translation by Naoki Abe.
14. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press (1996)
15. Koehn, P.: Europarl: A multilingual corpus for evaluation of machine translation. Draft (2002)
16. Ouimet, M.: Transsearch anglais-espagnol. `http://www.iro.umontreal.ca/~ouimema/ift3051/README.html` (2002)
17. Langlais, P., Carl, M., Streiter, O.: Experimenting with phrase-based statistical translation within the iwslt 2004 chinese-to-english shared translation task. In: International Workshop on Spoken Language Translation, Kytio, Japan (2004)
18. Koehn, P.: Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In: Meeting of the American Association for Machine Translation (AMTA), Washington DC (2004)