Parallel Corpus Refinement as an Outlier Detection Algorithm

Kaveh Taghipour and **Shahram Khadivi**
Human Language Technology Lab
Department of Computer Engineering and IT
Amirkabir University of Technology
{K.Taghipour, Khadivi}@aut.ac.ir

Jia Xu
DFKI GmbH, Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrücken Germany
Jia.Xu@dfki.de

Abstract

Filtering noisy parallel corpora or removing mistranslations out of training sets can improve the quality of a statistical machine translation. Discriminative methods for filtering the corpora such as a maximum entropy model, need properly labeled training data, which are usually unavailable. Generating all possible sentence pairs (the Cartesian product) to generate labeled data, produces an imbalanced training set, containing a few correct translations and thus inappropriate for training a classifier. In order to treat this problem effectively, unsupervised methods are utilized and the problem is modeled as an outlier detection procedure. The experiments show that a filtered corpus, results in an improved translation quality, even with some sentence pairs removed.

1 Introduction

Statistical machine translation systems need large parallel corpora to estimate the parameters of the translation models. Such a corpus can be built manually by human translators, but it is too costly and takes too much time. Various automatic methods for collecting parallel sentence pairs are proposed to reduce the time and energy needed to have a parallel corpus, but it comes at a cost. Automatically generated parallel corpora usually contain some noisy sentence pairs and are not perfect. Hence, the translation quality of a translation model, which is trained on such corpora, may not be satisfying.

Several methods of automatic generation of parallel corpora have been proposed by researchers. Some researchers have used alignment techniques to find sentence pairs. Resnik (1999) suggests a method to find an alignment between two sets of sentences in two different languages to build a corpus. Having a proper criterion for scoring candidate sentence pairs can improve the accuracy of the model. Some researchers use characters (Gale and Church, 1991) and words (Brown et al., 1991) to form a scoring function and then use it to find alignments. Iterative algorithms for sentence alignment are also suggested in some papers. As a sample of iterative algorithms, an EM-based unsupervised bilingual information extraction method is proposed in (Lee et al., 2010). The results of using a word-based translation model to find a proper alignment between two sets of sentence pairs are provided and discussed in various papers (Chen, 1993; Wu, 1994). Extracting fragments (gathering sub-sentences instead of full sentence pairs) to improve machine translation performance is also studied in some papers (Munteanu and Marcu, 2006).

After extracting parallel sentence pairs, it might be a good idea to reduce the noise level in a separate post-processing phase by another filtering model. The effect of different noise levels on the translation quality is studied in (Khadivi and Ney, 2005). The paper shows that the quality of the translation improves when the training corpus contains less noisy sentence pairs. Khadivi and Ney (2005) also remove noisy sentence pairs by scoring them based on a linear combination of two lexical models. Several researchers have studied the filtering problem and proposed some solutions. Sarikaya et al. (2009) and Turchi et al. (2009) use machine translation mod-

els to remove the noisy sentence pairs. They train a translation model on a noisy corpus and then use it in different ways to refine the training data. Filtering techniques based on the maximum entropy principle are also used in some studies. After gathering sentence pairs, Munteanu and Marcu (2005) use a maximum entropy model to detect and remove noisy pairs. Although we use a different model, some of the features used in (Munteanu and Marcu, 2005) are used in our model too. Tillmann (2009) uses a maximum entropy model as a filter and a beam search to extract sentence pairs.

The noise is usually a complex concept. It is hard to have a proper definition for noise, but it is relatively easy to have one for the parallel sentence pairs. Thus, a simple definition can be *'being not parallel'*. However, the definition is ambiguous and not precise since it presumes the noise as a discrete and binary concept. In this paper, we concern noise as a spectral characteristic of all sentence pairs and would like to remove sentences with higher degree of noise. Moreover, if all sentence pairs are assumed as vectors with different directions in space, parallel sentence pairs have approximately the same directions while noisy ones may have any arbitrary angle. This characteristic of noise makes it difficult to be modeled. Hence, using discriminative methods to classify sentence pairs as noise or parallel (Munteanu and Marcu, 2005) might be inappropriate. In order for a binary discriminative classifier to have a high power of generalization, it needs to see enough observations of both classes in the training set but it is hard to provide adequate samples of noise due to the variability of the noisy samples. Such a model would find it difficult to recognize unseen data with an unknown direction. Since the generation of such a labeled training set seems misleading, we will use unsupervised methods to handle the problem. The main idea is to learn the direction of the parallel samples and classify the most *novel* samples as noise by means of density estimators. Although, we assume that the number of noisy sentences are not too much, we show that the accuracy is acceptable even with a noise level as high as 30% or 40%. It has to be considered that in the case of density estimation, no weighting for features is needed.

In this paper, we use some unsupervised outlier detection methods to filter out noisy sentence pairs. The idea is to map the sentence pairs to an n-dimensional space and then remove the outliers. We show that omitting sentence pairs that are determined as outliers, reduces the noise of a corpus and thus improves the translation quality.

The second section of this paper briefly notices the outlier detection approaches that are used in this study. We will explain the features in the third section and the last part of the paper contains the experiments and results.

2 Outlier detection based on probability density estimation

In order to detect the outliers, we use density estimators and assume points with lower densities as outliers. Since presuming a well-known model as the generator for the data might be misleading, we use non-parametric methods for estimating densities. A good overview of the non-parametric density estimators can be found in (Silverman, 1986). The general expression for non-parametric density estimation is

$$\hat{f}(x) \simeq \frac{K}{nV} \tag{1}$$

where n is the total number of samples, V is the volume around x and k is the number of samples that fall in the volume V. Among different approaches to estimate probability density function, we try kernel-based techniques and also methods based on K-nearest neighbors. Kernel density estimators are one sort of methods that use the equation 1 by fixing V and varying K and K-nearest neighbor methods fix K and vary V. We selected kernel density estimators and K-nearest neighbor methods to study the problem in both cases. The methods are explained later in detail.

The methods explained here are similar to the famous algorithm DENCLUE (Hinneburg and Gabriel, 2007). This algorithm can be used to detect the outliers but it also tries to find the cluster boundaries too. In our case, we just try to detect the outliers, and cluster boundaries are of no use for us. After estimating density, we filter the data points with respect to the corresponding estimated density and remove the least dense points.

2.1 Density Estimation with Kernels

Given a sample of n observations $X_1, ..., X_n$ in a d-dimensional space, a kernel based density estimator with kernel K is defined by (Silverman, 1986)

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \quad (2)$$

where h is the window width or bandwidth and the kernel K satisfies the condition

$$\int_{-\infty}^{\infty} K(x)d(x) = 1 \quad (3)$$

Different kernels can be used to estimate the density. The kernels that are used in this study are the *Epanechnikov*, *Gaussian* and *Laplace* kernels (Li and Racine, 2007; Li and Ruppert, 2008).

The Epanechnikov kernel is defined by

$$K(x) = \frac{3}{4}(1 - x^2)\mathbf{1}_{\{|x| \leq 1\}} \quad (4)$$

where the $\mathbf{1}_{\{...\}}$ is the indicator function. The indicator function in the Epanechnikov kernel formulation forces the output to be zero when the value for x, exceeds the boundary.

The definition for the Gaussian kernel is

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad (5)$$

This kernel function is not truncated and might be better when the bandwidth is small, but it is slower than the Epanechnikov kernel.

Finally, the Laplace (or double-exponential) kernel is defined by

$$K(x) = \frac{1}{2} exp(-|x|) \quad (6)$$

The Laplace kernel has the heaviest tail among the kernels used in the experiments. A heavy tail of a kernel makes it suitable when the training and test data differ a lot or an unsuitable bandwidth is selected.

Small values for the bandwidth makes the estimation more detailed, while choosing larger values results in a smooth function. Due to the importance of this parameter, we use automatic methods to find a suitable bandwidth. Among various automatic bandwidth selection methods, we use the plug-in selectors (Sheather and Jones, 1991), which try to minimize an error function such as mean integrated squared error (MISE) or asymptotic MISE (AMISE). The definitions for MISE and AMISE are presented below:

$$MISE(h) = E \int (\hat{f}_h(x) - f(x))^2 dx \quad (7)$$

$$AMISE(h) = \frac{R(K)}{nh} + \frac{1}{4} m_2(K)^2 h^4 R(f'') \quad (8)$$

where h is the bandwidth, f'' is the second derivative of f and if $g(x)$ is a function then

$$R(g) = \int g(x)^2 dx \quad (9)$$

and

$$m_2(K) = \int x^2 K(x) dx \quad (10)$$

Variable bandwidth kernel methods are estimators, which use variable or adaptive bandwidth for density estimation. We also try the variable bandwidth kernel estimator of (Kim and Scott, 1992) to estimate the density.

2.2 Density Estimation with Nearest Neighbor Methods

It is possible to estimate the probability density function by the well-known K-nearest neighbor methods. In this case, the degree of smoothing is adapted to the local density of given sample of data and is controlled by the parameter K (Silverman, 1986). Given a sample of n observations $X_1, ..., X_n$, a distance function $d(x, y)$ between two points x and y on the line is $|x-y|$, and for each t, $d_k(t)$ is defined such that

$$d_1(t) < d_2(t) < ... < d_n(t) \quad (11)$$

in which the distances are sorted ascending. Then the K-nearest neighbor estimator can be defined by

$$\hat{f}(x) = \frac{K}{2nd_k(x)} \quad (12)$$

The choice of parameter K is important, because it specifies the smoothing degree. Usually the value for K is selected around $\sqrt{n}$.

3 Features

The feature set used in this paper is a combination of novel features and previously suggested ones. The features can be categorized into 4 groups:

- Translation model probabilities
- Word alignments
- N-gram language model
- Sentence length

It is also worth mentioning that all features are trained on the noisy training set. The reason is to preserve the independence of the approach from other sources of clean or labeled data. Although training basic models on the noisy data might lead to unacceptable feature functions, we believe that the models are robust enough to maintain their functionality when the degree of noise is low, relative to the parallel sentence pairs. Additionally, the features complement and supplement each other since they consider different aspects of the complex noise concept. The features are designed in a way that distribute the noisy samples, while concentrating the parallel ones in a compact part of space. Also, to have a homogeneous set of features, we normalize all feature functions to vary at the boundary of [0 1]. Finally, the feature selection procedure is explained in section 4.

3.1 Translation model probabilities

We train the word-based translation model IBM-1 (Brown et al., 1993) on the noisy corpus and use the conditional translation probabilities as a feature. IBM model 1 is defined by

$$P_{IBM1}(e|f) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \quad (13)$$

where f and e are source and target sentences respectively and $t(f_j|e_i)$ is the probability of translating the word f_j into e_i. We use the IBM model 1 to compute five features, which are listed below:

- $P_{IBM1}(e|f)$
- $P_{IBM1}(f|e)$
- Unnormalized $P_{IBM1}(e|f)$
- Unnormalized $P_{IBM1}(f|e)$
- Average of $P_{IBM1}(e|f)$ and $P_{IBM1}(f|e)$

The unnormalized features are the outputs of the model without the normalizing factors. Similar features are used in several papers such as (Khadivi and Ney, 2005). Since the goal is to improve the quality of a statistical machine translation, it seems useful to use translation model probabilities as feature functions. Among different translation models, word based models can be utilized much easier because of their simplicity. The reason of using IBM model 1 is that this model is simpler and faster and produces a sufficiently reliable measure of literal translation. However, building other feature functions based on more complex translation models can be useful to detect and remove mistranslations.

3.2 Word alignments

Features based on word alignments are used in some papers (Munteanu and Marcu, 2005). In order to get a word alignment between two sentences, we use the noisy corpus to train the IBM models (Brown et al., 1993) in both directions and as a result we get the source-to-target and target-to-source word alignments. By using the heuristics suggested in (Och and Ney, 2003), a symmetric word alignment is produced. We use the word alignments to calculate features:

- Number and percentage of null alignments in both source and target sentences
- Number and percentage of null alignments in the source sentence
- Number and percentage of null alignments in the target sentence
- Alignment entropy of both source and target sentences
- Alignment entropy of the source sentence
- Alignment entropy of the target sentence

Null alignments are the words which are not aligned to any words of the other sentence and the

alignment entropy is a criterion of the alignments distribution. This feature scores the uniform alignments with a higher value. The source and target alignment entropies are defined by

$$ent(e_1^I, A) = \frac{\sum_i q(e_i, A).log(q(e_i, A))}{log(I)} \quad (14)$$

and

$$ent(f_1^J, A) = \frac{\sum_j q(f_j, A).log(q(f_j, A))}{log(J)} \quad (15)$$

where $q(x_k, A)$ is a normalized number of links in which x_k is involved. Finally, we define the total alignment entropy by

$$ent(e_1^I, f_1^J, A) = ent(e_1^I, A).ent(f_1^J, A) \quad (16)$$

This feature tends to score alignments with higher degree of uniformity with larger values. The idea is that when two sentences are translations of each other, usually the alignments follow an approximately uniform distribution. But when a sentence pair is a noisy entry, the word alignments are mostly seen on separated parts of a pair.

3.3 N-gram language model

The next group of feature functions are based on the language model probabilities of the source and target sentence. We use M-gram language models to estimate the sentence probabilities. Given a sequence of words $w_1, ..., w_n$, the language model probability of the sequence is defined by

$$P_{LM}(w_1^n) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \quad (17)$$

Usually, the length of the history w_1^{n-1} is limited, to have a feasible language model. Hence, an M-gram language model is defined by

$$P_{LM}(w_1^n) = \prod_{n=1}^{N} p(w_n|w_{n-M+1}^{n-1}) \quad (18)$$

The language model based features used in the model are:

- Source and target 3-gram probabilities
- Difference and ratio of the source and target 3-gram probabilities

	Train	
	En	Fr
#Sentences	45083	
#Words	330154	348483
Avg. #Words	7.3	7.7

Table 1: Corpus Statistics: Training set

	Dev.		Test	
	En	Fr	En	Fr
#Sentences	1000		1000	
#Words	27620	29999	27594	30159
#OOVs	795	1151	800	1148

Table 2: Corpus Statistics: Dev. and Test sets

3.4 Sentence length

The last group of feature functions are based on the length of a sentence pair. These features are used in many papers to detect mistranslations (Munteanu and Marcu, 2005; Gale and Church, 1991; Brown et al., 1991). Following these papers, we use the sentence length based on both characters and words to form new features for the model. The feature functions are the difference and ratio of the source and target lengths based on words and characters. By the help of the sentence length, it is easy to detect mistranslations of large difference in source and target lengths.

4 Experiments

We use the proposed approach to filter an automatically generated French-English parallel corpus, in which the sub-sentence pairs are gathered from the multilingual Euronews website[1]. In order to evaluate the model, we use the filtered corpus as the training data for a phrase-based statistical translation model (Koehn et al., 2003). The test and development sets are selected from the parallel Europarl corpus[2]. Some statistics about the data sets is provided in the tables 1 and 2.

Here, we use the rule of thumb (ROT) plug-in method for a fast and appropriate bandwidth selection. ROT minimizes the AMISE criterion and is

[1] www.euronews.net

[2] www.statmt.org/europarl/

	Dev.	Test
$P_{IBM1}(e\|f)$	33.62	33.12
$P_{IBM1}(f\|fe$	33.54	33.07
Unnormalized $P_{IBM1}(e\|f)$	33.27	32.97
Unnormalized $P_{IBM1}(f\|e)$	33.31	32.91
$Avg(P_{IBM1}(e\|f), P_{IBM1}(f\|e)$	33.49	33.46

Table 3: Evaluating translation model features: BLEU score

	Dev.	Test
#null alignments (source)	33.57	32.94
#null alignments (target)	33.76	33.12
#null alignments (total)	33.71	33.17
%null alignments (source)	33.73	33.08
%null alignments (target)	33.76	33.36
%null alignments (total)	33.68	33.23
Alignment entropy (source)	33.41	33.00
Alignment entropy (target)	33.22	32.91
Alignment entropy (total)	33.05	32.86

Table 4: Evaluating word alignment features: BLEU score

much faster than other methods. Using other methods for bandwidth selection can effect the estimation, but are not tested here due to their less importance.

In order to have a proper feature set, we evaluate each feature independently. The evaluation is done by removing each feature, re-estimating the densities and the observing the translation quality (BLEU). This method of feature selection may not be a perfect solution because it depends significantly on the tuning set. However, considerable decrease in BLEU score, may be a sign of degrader feature function. The estimation method that is used in feature evaluation is the kernel-based density estimator with the Gaussian kernel exploited. As mentioned before the ROT selector is used to adjust the bandwidth. The tables 3, 4, 5 and 6 display the effect of each feature function on the translation quality.

Automatically generated corpora may contain different levels of noise. In order to estimate the noise level of the target corpus, translation quality (BLEU) is illustrated against the percentage of filtered sentence pairs in figure 1. The plot shows that trim-

	Dev.	Test
$P_{LM}(source)$	33.62	33.12
$P_{LM}(target)$	33.78	33.24
$P_{LM}(source) - P_{LM}(target)$	33.54	33.42
$P_{LM}(source)/P_{LM}(target)$	33.72	33.09

Table 5: Evaluating language model features: BLEU score

	Dev.	Test
Diff. of lengths (words)	33.29	33.30
Ratio of lengths (words)	33.45	33.09
Diff. of lengths (characters)	33.71	33.21
Ratio of lengths (characters)	33.55	33.17

Table 6: Evaluating length-based features: BLEU score

ming 5% of the corpus leads to a relatively improved translation quality. As expected, removing higher number of sentence pairs (e.g. 50%) from training data results in decreased translation quality, but an interesting point is that with higher levels of trimming, translation quality reduces with much lower pace. Even with 50% filtering and training the SMT with half of the training set, the translation quality only reduces 0.6 BLEU (0.02 relative to the baseline). 50% of filtering means a model with approximately half the size of the baseline and thus less memory needs.

We remove a fixed percentage of data, instead of using a fixed threshold for density. After removing 5% of sentence pairs that are detected as the least dense points, the filtered corpus is used as the training data for a phrase-based machine translation. The results for different kernel-based methods are displayed in the table 8 and KNN-based methods in 7. Each row of the table shows the translation quality in terms of BLEU (Papineni et al., 2002) for different estimators.

It has to be mentioned that the baseline is trained on the full corpus, and other rows of the table are the models with 5% fewer sentence pairs. All estimators are dependent and sensitive to the bandwidth parameter.

Two tests are performed to prove that the improvement is statistically significant. We have utilized the bootstrap resampling method to resam-

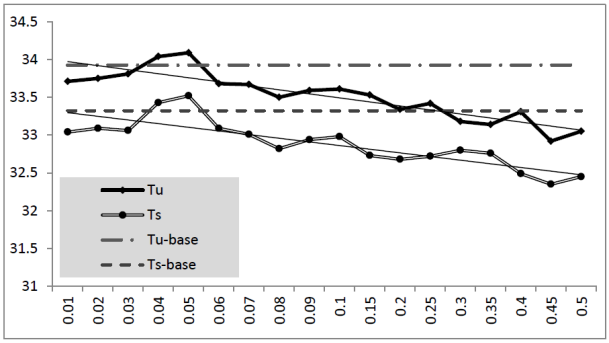

Figure 1: Results (BLEU) of evaluating the SMT systems for Dev. (*Tu*) and test sets (*Ts*). The X-axis shows the percentage of sentence pairs removed from the training corpus. The baseline scores are illustrated by *Tu-base* and *Ts-base*

	Dev.	Test
Baseline	33.93	33.34
$K = 50$	33.67	32.94
$K = 100$	33.79	33.06
$K = 150$	33.82	33.21
$K = 200$	33.97	33.28
$K = 250$	33.85	33.17
$K = 300$	33.86	33.12

Table 7: Translation quality (BLEU) on the test and dev. sets when filtered with KNN estimators

	Dev.	Test
Baseline	33.93	33.34
AGF	33.98	33.09
Epanechnikov	**34.01**	**33.51**
Gaussian	33.79	33.30
Laplace	33.72	33.22
KNN	33.89	33.24

Table 8: Translation quality (BLEU) on the test and dev. sets when filtered with kernel estimators

	Dev.		Test	
	WSR	T	WSR	T
5%, Epan., ROT	1e-64	0	5e-12	1e-12

Table 9: P-Values obtained from Wilcoxon signed-rank test (WSR) and T-test (T) performed on samples of bootstrap resampling method. The target system is the SMT trained on 95% of training data, which is filtered by Epanechnikov kernel and the ROT bandwidth selection method

ple the development and the test sets and then use Wilcoxon signed-rank (Wilcoxon, 1945) and the T-test to evaluate the amount of improvement. The results (P-values) are shown in the table 9. It has be noticed that the Wilcoxon test is performed directly on BLEU scores of two SMT systems, but the T-test is used on the differences of two BLEU scores.

5 Conclusion

In order to detect and remove the mistranslations in a parallel corpus, we mapped each sentence pair into an N-dimensional feature space and then estimated the density for each one of them. The least dense points are treated as outliers and thus are removed from the corpus. After filtering the corpus, we used it to train a phrase-based machine translation system and evaluated the method.

The results show that the filtered corpus with fewer sentence pairs not only results in comparable quality to the larger corpus, but also improves the quality when a proper estimator is used. The results prove our claim that the model selects better translations and does not remove the useful phrase pairs. Having a larger test set, in which most of the training phrase pairs are used, the improvement would be much higher. We also showed that the method can be used to reduce the translation model size with a minimum cost of quality degradation. Thus the method can be utilized to reduce the model size and thus less memory consumption with a trade-off between the translation quality and memory needs.

All the features used in this study are evaluated against translation quality. We removed each feature and re-estimated the points to make sure that none of the feature functions are reducing the quality on the tuning set. However, having a better feature set would definitely improve the efficiency. Although it involves supervision, we also plan to use clean parallel corpora to have a more precise detection algorithm and also a better feature selection method by finding better feature functions.

Acknowledgments

This work was partially supported by Research Institute for ICT (ITRC) contract No. 500/1141.

References

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, ACL '91, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June.

Stanley F. Chen. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

William A. Gale and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, ACL '91, pages 177–184, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander Hinneburg and Hans-Henning Gabriel. 2007. Denclue 2.0: Fast clustering based on kernel density estimation. In *In Proceedings of the 7th International Symposium on Intelligent Data Analysis*, pages 70–80.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In *Proceedings 10th International Conference on Application of Natural Language to Information Systems, NLDB 2005*, pages 263–274. Springer Verlag.

Jooseuk Kim and Clayton Scott. 1992. Variable kernel density estimation. *Annals of Statistics*, 20:1236–1265.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lianhau Lee, Aiti Aw, Min Zhang, and Haizhou Li. 2010. Em-based hybrid model for bilingual terminology extraction from comparable corpora. In *Coling 2010: Posters*, pages 639–646, Beijing, China, August. Coling 2010 Organizing Committee.

Q. Li and J.S. Racine. 2007. *Nonparametric econometrics: Theory and practice*. Princeton University Press Princeton, NJ:.

Y. Li and D. Ruppert. 2008. On the asymptotics of penalized splines. *Biometrika*, 95(2):415.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31:477–504, December.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 81–88, Sydney, Australia, July. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29:19–51, March.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 527–534, College Park, Maryland, USA, June. Association for Computational Linguistics.

R. Sarikaya, S. Maskey, R. Zhang, E. E Jan, D. Wang, B. Ramabhadran, and S. Roukos. 2009. Iterative Sentence-Pair extraction from Quasi-Parallel corpora for machine translation. In *Tenth Annual Conference of the International Speech Communication Association*.

S. J. Sheather and M. C. Jones. 1991. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):pp. 683–690.

B. W. Silverman. 1986. *Density estimation for statistics and data analysis*. Chapman and Hall.

Christoph Tillmann. 2009. A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 225–228, Suntec, Singapore, August. Association for Computational Linguistics.

Marco Turchi, Tijl De Bie, and Nello Cristianini. 2009. An intelligent agent that autonomously learns how to translate. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '09, pages 12–19, Washington, DC, USA. IEEE Computer Society.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.

Dekai Wu. 1994. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 80–87, Stroudsburg, PA, USA. Association for Computational Linguistics.