Unsupervised Bilingual Segmentation using MDL for Machine Translation

Bin Shan, Hao Wang, Yves Lepage
Graduate School of Information, Production and Systems
Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
{reynolds@fuji., oko_ips@ruri., yves.lepage@}waseda.jp

Abstract

In statistical machine translation systems, a problem arises from the weak performance in alignment due to differences in word form or granularity across different languages. To address this problem, in this paper, we propose a unsupervised bilingual segmentation method using the minimum description length (MDL) principle. Our work aims at improving translation quality using a proper segmentation model (lexicon). For generating bilingual lexica, we implement a heuristic and iterative algorithm. Each entry in this bilingual lexicon is required to hold a proper length and the ability to fit the data well. The results show that this bilingual segmentation significantly improved the translation quality on the Chinese–Japanese and Japanese–Chinese subtasks.

1 Introduction

Words are generally the smallest processing units in varieties of NLP tasks. However, there is no guarantee that such smallest processing units can fit any NLP tasks. Especially in bilingual tasks (e.g. statistical machine translation), different languages have different writing systems or segmentation granularity. Such problem should be considered as a critical factor of performance in translation quality. For instance, in machine translation experiments on 11 Europarl corpora (Koehn, 2005), Finnish has the lowest translation accuracy as evaluated by BLEU scores when translated into English. French–Spanish has the highest BLEU scores. Finnish is a non-Indo-European and agglutinative language. French and Spanish have very similar grammar. Thus, the problem arising from different grammatical structure could lead a poor generalization when training SMT system uses such data. This is one aspect. Another aspect, there still exists some problem even segmenting language to generate similar vocabulary. In our view, we suppose that similar units should have a proper size. If similar units are too general, it will cause that size of model become too large and a over-fitting problem in model itself. Namely, too general similar units could not solve this problem indeed. Too general similar units problem also appears in (Virpioja et al., 2007) where they perform monolingual segmentation at the morphological level for Finnish-English translation and put the segmented data to a phrase-based statistical machine translation system. That paper indicates the segmented corpus has lower out-of-vocabulary rates and generates more refined phrases with better generalization ability. However, the results of experiment show that they could not improve translation accuracy. In their method, the sentences already had similar units by morphological level segmentation. However, as we mentioned earlier, over-general similar units also go against on improving the translation quality.

On account of those problem, we suppose that data should be segmented through more proper method which could generate similar units holding proper size and goodness-to-fitting data. Fortunately, minimum description length (MDL) principle as an important principle in information theory has shown a good performance in finding units which could hold a trade-off on that aspect. More details about this

31st Pacific Asia Conference on Language, Information and Computation (PACLIC 31), pages 89–96
Cebu City, Philippines, November 16-18, 2017

technology are discussed in section 2.1.
In this paper, we firstly introduce the main technology. Then we propose a bilingual model and an iterative search algorithm to generate the best model. To evaluate our approach, we put the segmented corpus by our method into Moses (Koehn et al., 2007) and use BLEU score and NIST score as an evaluated measure.

2 MDL-based segmentation

2.1 Minimum description length

The Minimum Description Length was first introduced by (Rissanen, 1978). In our method, we suppose to use Crude MDL (Grünwald, 2005), which has two parts.

$$\begin{aligned} M' &= \arg\min_{M} \mathrm{DL}(D, M) \\ &= \arg\min_{M} \mathrm{DL}(M) + \mathrm{DL}(D|M) \end{aligned} \tag{1}$$

Where $\mathrm{DL}(.)$ denotes the description length. The $\mathrm{DL}(D|M)$ represents the description length of data given by model or data cost. $\mathrm{DL}(M)$ is the description length of the model or model cost. The principle requires a minimum model, which can produce a lowest description length of two parts. The $\mathrm{DL}(D|M)$ requires that the model has better ability to fit the data. The $\mathrm{DL}(M)$ requires that the model has simpler structure. As González-Rubio and Casacuberta (2015) said, the MDL provides a joint estimation of the structure and parameters (probability distribution) of the model. It naturally provides a mechanism against over-fitting or being too general by implementing two parts in this principle.

2.2 Related works

MDL has been used in common inductive inference tasks (Grünwald, 2005). In this section, we mainly introduce applications. De Marcken (1996) tries to infer the monolingual grammar structure using MDL. Yu (2000) introduce unsupervised monolingual word induction approach using MDL. Approximately, Hewlett and Cohen (2011) implement a heuristic search algorithm and use MDL as criterion to produce the best monolingual segmentation scheme. Zhikov et al. (2010) also employ an MDL-based as criterion with a more efficient greedy algorithm. Chen (2013) proposes a compression-based method using MDL and improve the performance of monolingual segmentation. Argamon et al. (2004) use an efficient recursive method on morphological segmentation using MDL. Those early works focus on exploiting MDL to achieve monolingual segmentation, and indicate that MDL-based method has an excellent performance on unsupervised monolingual segmentation. For bilingual NLP tasks using MDL, Saers et al. (2013) try to build an inversion transduction grammars with MDL. González-Rubio and Casacuberta (2015) try to improve the translation quality by inferring a phrase-based model using MDL. Actually, those works focus on achieving different NLP tasks using MDL.
Our work employs the same technologies as previous works. However, we extend MDL-based monolingual model to bilingual. In addition, previous works using MDL on bilingual tasks did not give the bilingual segmentation method. However, we focus on simultaneously segmenting bilingual data.

3 Methodology

3.1 Bilingual model

Our method builds a bilingual word segmentation scheme. Comparing with the monolingual models, we propose the bilingual model. The bilingual model M can be represented as a bilingual lexicon (a set of unit pairs).

$$M = \{a_i \mid a_i = (s_i, t_i), s_i \in S, t_i \in T\}$$

(s_i, t_i) is the ith unit pair in M, and S and T respectively belongs to source and target types sets. s_i and t_i are source units and target units. Moreover, a single symbol is a basic unit in the monolingual setting. For the bilingual setting, we could extend to choose single symbol pairs as basic units. Thus, if the set only consisting of basic units, we call it basic set M_{basic}. Figure 1 illustrates the similarities and differences between units in the monolingual and bilingual. there are varieties of interpretations to MDL-model using different technologies. Our formula mainly is derived from Zhikov et al. (2010) and Yu (2000).
Generally, the description length of data given by

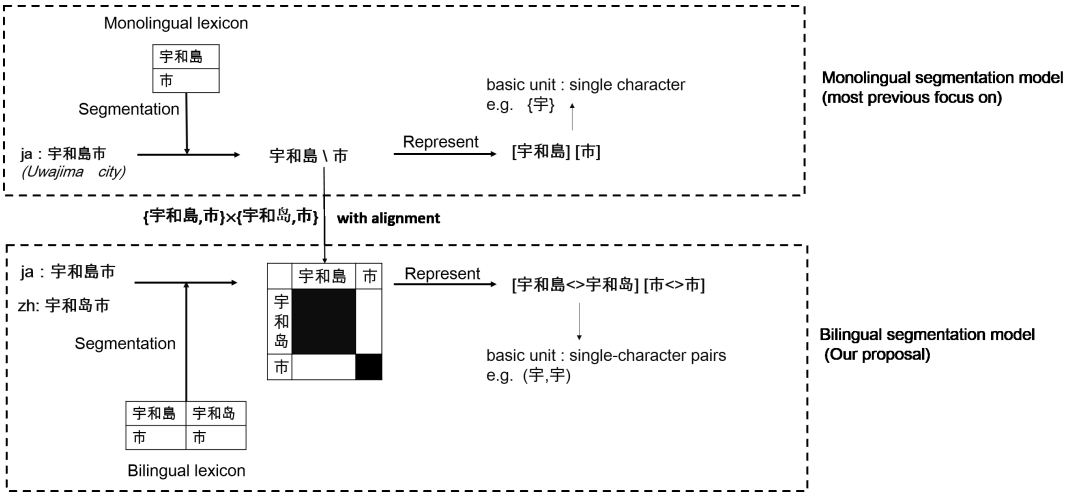


Figure 1: Monolingual and Bilingual

1. The essence of bilingual model is treated **the Cartesian product** as the set of source and target types **with alignment**.
2. A basic unit in a monolingual sentence is a single character / letter, which in a bilingual sentence should be a single character/letter pair.
3. Any sentences can be represented as several units following the order according to the monolingual / bilingual lexicon. For representation, " [...] " represents a unit. "... <> ..." represents an alignment which is used to connect the source and target word.

model DL$(D|M)$ is calculated using Shannon-Fano code. For the data cost,

$$\text{DL}(D|M) = \sum_{i}^{M} -C(a_i) \log P(a_i) \qquad (2)$$

Where $P(a_i) = -\log \frac{C(a_i)}{N}$ is the *self-information* of a_i. a_i represents an alignment unit (s_i, t_i). $C(a_i)$ is a frequency of a_i in data D. Equation 2 gives the total information contained in the data given by the model M.

For the description length of model DL(M), different work pieces introduce different calculations. The common point in the calculation is the product of the length in character of units and an estimate of per-character entropy (Zhikov et al., 2010) (in the bilingual setting, "character" should be replaced with "character pairs" or "basic unit pairs"). The estimate of every basic unit pairs entropy is not easy, Yu (2000) suggests to use average entropy as estimation. Using average entropy as estimation will improve the speed of implementing our following algorithm a lot. Namely, the calculation of model cost generally covert to count the size of model. However, with this estimation, we could not capture the probability distribution of basic units. Thus, at the precision perspective, we ignore the effects of substructure. So we calculate model cost using

$$\text{DL}(M) = \sum_{i}^{|M|} b \times len(a_i) \qquad (3)$$

Where $len(a_i)$ is the number of basic alignment units in a_i. $b = -\log_2 |M_{ini}|$ and which represents binary code length of initial model. Where M_{ini} is the simplest bilingual lexicon (model) which has the lowest model cost and just includes basic unit pairs. $len(M_{ini})$ is the basic lexicon size. Thus, b is constant when the data given.

For the basic model M_{ini} , it should have the lowest description length of the model. Besides, it is an initial model in our method. However, the description length of data given by the initial model in most cases will be very large. So we need to merge some smaller unit pairs into some bigger ones in order to decrease the description length of data. Likewise, the description length of the model will increase if we merge some unit pairs. Therefore, there exists a trade-off in two parts and the best model we excepted is such a trade-off model.

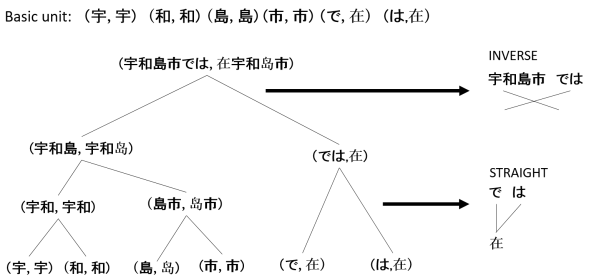


Figure 2: A efficient searching path by ΔDL

1. In this binary tree, the leaves are the basic unit. Every node is an alignment unit. Every father node can be represented by the child node.
2. Input the candidates can be represented as two child nodes.
3. Two child nodes should be combined into a father node with two ways: INVERSE and STRAIGHT.

3.2 Bilingual segmentation

As De Marcken (1996) showed, every sentence has a hierarchical structure and he calls the Viterbi representation for a sentence. He tries to search the best model by inputting possible candidates with two operations (add and delete). They represent candidates as a binary combination of two units which could be found in the current model. Likewise, Lardilleux et al. (2012) shows how to segment bilingual sentences by building the bilingual binary tree structure with a recursive binary splitting method. The same place in previous works, they all choose a binary combination or split way to search the best model. Actually, this measure is a common way to search the best model by using MDL principle. The binary representation brings an efficient path to search the best model. We just evaluate the changes in description length, when possible candidates are applied to the current model.

So our problems can be converted to evaluate the changes in description length after a new alignment unit is accepted by model. Every accepted candidate will bring a ΔDL, it can search the best model by evaluating the changes (Figure 3.2). Another important point, from those structures we can find that there exist two direction search algorithms. Those are bottom-to-top search method with binary combination and top-to-bottom with binary split.

3.3 Quantifying changes in description length

The MDL-based method provide an evidence to define the best model with the sum of data and model cost. Our method employs a heuristic algorithm to iteratively generate a new model from the current model. Due to our model is bilingual lexicon, we generate new model through adding possible candidates to current lexicon. For giving the evidence of possible candidates, every candidate should be evaluated to a change ΔDL in description length. When the ΔDL can decrease the DL(D, M), the candidates will be applied to the current model. For example, when we apply a candidate a_1a_2, it can be represented as a_1 and a_2 in current model M.

Considering the MDL-based methods generally consist of model and data cost, the changes are evaluated as:

$$\Delta\text{DL}(D, M) = \Delta\text{DL}(M) + \Delta\text{DL}(D|M)$$

For a candidate a_1a_2 to be feed into the model, we just evaluate the changes of two parts.

For the ΔDL$(D|M)$ with four parts:

$$\Delta DL(D|M) = \delta_1 + \delta_2 - \delta_3 + \delta_4$$

$\delta_1 = (C(a_1) - C(a_1a_2)) \log \frac{C(a_1) - C(a_1a_2)}{N - C(a_1a_2)}$ is difference on a_1,

$\delta_2 = (C(a_2) - C(a_1a_2)) \log \frac{C(a_2) - C(a_1a_2)}{N - C(a_1a_2)}$ is difference on a_2,

$\delta_3 = C(a_1a_2) \log \frac{C(a_1a_2)}{N - C(a_1a_2)}$ is difference on new input a_1a_2,

$\delta_4 = K \log \frac{N'}{N}$ are changes on other alignment units, actually we can find the changes on other alignment units just are about the total number. K is the number of other alignment units.

For the ΔDL(M),

$$\Delta\text{DL}(M) = b \log \frac{len(a_1) + len(a_2)}{len(a_1a_2)} = b\delta_m$$

As shown in the above formula, b is a constant and we just need to focus on changes of the total model length. As for changes on length of model, we just need to care about whether any inputs change the counts of old units in model to 0. Due to the counts change into 0, it should be removed from the model. We assign $\frac{len(a_1)+len(a_2)}{len(a_1a_2)}$ as the difference value δ_m. So we have:

1. When frequency of a_1 or a_2 changes to 0 after input operation, the $\delta_m = 1$

2. When frequency of a_1 and a_2 changes to 0 after input operation, the $\delta_m = 0$

3. When frequency of a_1 and a_2 does not change in 0 after input operation, the $\delta_m = 2$

By calculating the sum of changes on two parts, we can give the inputs an evidence about accepting or not.

3.4 Search Algorithm

The previous section introduced that we use the ΔDL to evaluate changes of possible candidates on description length. However, the order of applying a new alignment unit is also very important. González-Rubio and Casacuberta (2015) introduced that the order of inputting candidates should be sorted by the ascending of $\Delta\text{DL}(D|M)$. In our method, we take the following strategy:

1. Segment corpus to characters and use word alignment tools to get a character alignment result as basic model.

2. Collect all the possible binary combination candidates from the data and model.

3. Run an iterative procedure to generate models.

4. Repeat the 2 to 3 until the description length will not reduce.

Algorithm 1 describes the processing of iterative generating model in step 3. First, we collect all possible candidates (line 2 to 3). Then we estimate the variation in description length when those candidates are applied to model (line 4 to 9). Then we evaluate the changes in total description length and use those candidates to update the model (line 11 to 15). Finally, the whole loop will end until the description length of the model could not reduce any more (line 17).

Algorithm 1 Iterative Generate Model

Input: M : Initial model consist of basic units
Output: M' : Generated model

```
1:  while Δ > 0 do
2:      Φ ← collect(D, M)
3:      candidates ← ascending_sort(Φ)
4:      for s ∈ candidates do
5:          delta = eval_DL_data(s)
6:          if delta > 0 then
7:              true_candidates.append(s)
8:          end if
9:      end for
10:     C ← ascending_sort(true_candidates)
11:     for s ∈ C do
12:         true_delta ← eval_total_DL(s)
13:         if true_delta > 0 then
14:             M' ← update(M, s)
15:         end if
16:     end for
17: end while
```

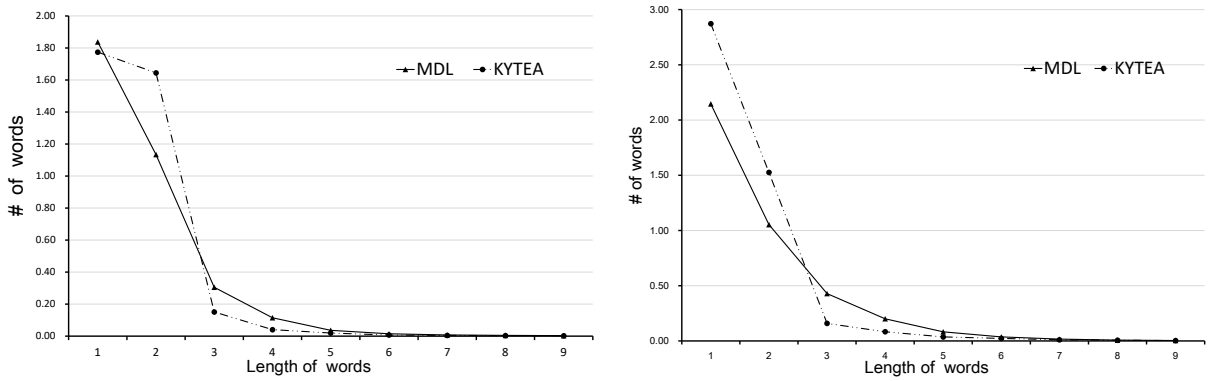
4 Experiment

Our method are evaluated through building Chinese–Japanese SMT experiments. For getting initial bilingual model, the extra alignment tool is used. The results obtained with the proposed method are compared the results obtained using *Kytea*[1] as segmentation technologies.

4.1 Setup

In our experiment, we use ASPEC [2] as experiment corpus. Due to the low performance of the current word alignment tools for character alignment on Latin languages, we cannot perform our method with the letter to letter alignment on Latin languages. However, it works well for Chinese and Japanese. So we select the Chinese and Japanese as our experiment corpus. For word alignment tools, we use *MGiza++*[3] to get character-based alignment results. To avoid unnecessary processing (e.g. resulted from non-Chinese units in Chinese corpus), we in advance token the non-Chinese or non-Japanese letter and as one unit. For machine translation system, we use *Moses*[4]. To benchmark our method, we choose data segmented by *Kytea* as baseline. The reason we choose Kytea is that it always segments the corpus with a small degree (the most cases are morpholog-

[1] http://www.phontron.com/kytea/
[2] http://orchid.kuee.kyoto-u.ac.jp/ASPEC/
[3] https://github.com/moses-smt/mgiza
[4] http://www.statmt.org/moses/



(a) Chinese (b) Japanese

Figure 3: Frequency and length of words in corpus segmented by MDL and Kytea

1. Kytea (monolingual segmentation method) have different granularity of the segmentation in Chinese and Japanese. However, bilingual MDL-based method **share similar granularity** across both languages.

2. Words segmented by Kytea have small granularity. However, our method (MDL-based segmentation) have **smoother distribution and larger segmentation granularity.**

ical level). We suppose it could show the unbalance problem in Chinese and Japanese more clearly. Table 1 illustrates the data setting of SMT experiment.

4.2 Result and analysis

The total number of iterations of our algorithm are 8 times. Figure 4 illustrates changes of each iteration in data cost, model cost and total cost. We found that MDL principle provides any candidates an evidence through introducing a change in two parts cost. MDL principle would find a best balanced cost of model and data. Figure 3 illustrates the frequency distribution of different length of words. The granularity of segmentation given by Kytea and our method is different, and our method assign a smoother frequency distribution than kytea. We also can found such phenomenon shown in data setting of SMT system (Table 1). In Table 1, we found average of length of words segmented by our method is longer than Kytea.

Due to different segmentation standards, we need to unify them in the evaluation step. Here, we evaluate translation accuracy in characters. Likewise, non-Chinese and non-Japanese are tokenized as one unit. Table 4.2 shows that the BLEU (Papineni et al., 2002) scores have improved **2.01%** in Chinese to Japanese. For NIST (Doddington, 2002) scores, we found that there are improvements in both translating directions.

5 Conclusion and Future Work

5.1 Conclusion

We propose a bilingual segmentation method using MDL, which aims at improving translation quality. Our method could simultaneously segment bilingual corpus and generates corresponding bilingual lexicon. Thus, our work also can be treated as a bilingual lexicon induction. Since our segmentation method achieves a slightly better translation result shown in Table 4.2, we conclude that our bilingual MDL-based segmentation method is more effective than previous monolingual segmentation method. Besides, we also found that MDL-based method could give more balanced trade-off between segmentation granularity and frequency. Differ with previous works using MDL-based method on monolingual segmentation, we extended the MDL-based method into bilingual segmentation and improved translation quality.

Our contributions in this work can be summa-

Data	Seg.	Sent.	Chinese		Japanese	
			Tokens	Length	Tokens	Length
Train	Kytea	135.0 k	3.66 M	10.82	4.74 M	11.33
	MDL		3.46 M	11.82	3.98 M	12.27
Tune	Kytea	3.0 k	84.1k	7.71	108.1 k	8.28
	MDL		79.4 k	8.36	90.4 k	9.03
Test	Kytea	11.0 k	308.4 k	8.94	396.6 k	9.47
	MDL		290.9 k	9.44	331.1 k	10.06

Table 1: Data setting
Length: average length of types in corpus;
Tokens.: number of word tokens in corpus;
Sent.: number of sentences in corpus;

	Seg.	BLEU	p-value	NIST	p-value
ja-zh	Kytea	36.68±0.28	<0.01	9.84±0.03	<0.01
	MDL	**38.69±0.28**		**10.24±0.04**	
zh-ja	Kytea	40.46±0.28	0.1	9.81±0.03	<0.01
	MDL	**40.35±0.28**		**10.08±0.03**	

Table 2: Experiment result

1. BLEU and NIST: translation accuracy metrics (based on characters)
2. p-value < 0.05 means the improvements are statistically significant different.

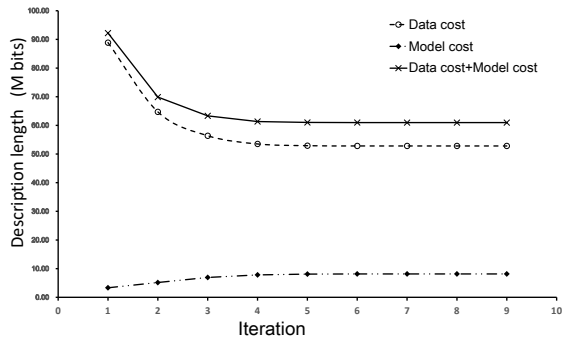


Figure 4: The data and model cost with iteration

rized as in three folds. Firstly, we propose a bilingual segmentation method instead of the monolingual method as an initial step of machine translation. Secondly, we choose MDL as main technology in our segmentation. This technology could be prone to produce more balanced word pairs in segmentation and gives a better inference on bilingual lexicon. Thirdly, our method is an unsupervised method based on characters which is also can be applied to any other languages writing in CJK characters.

5.2 Future Work

For languages written with the Latin alphabet, the basic unit is very limited. The current alignment tools will filter a large amount of characters alignment results. Thus, the bottom-to-top method cannot be applied. As mentioned in Section 3, there is also another strategies (top-down) which can be used to solve the problem. It will be in our future work. In addition, the initial model in our method depends on character-based alignment results. The quality of character-based word alignments is an influential factor in our final segmentation. A better method could be generate the initial model without any alignment tool. This could lead to better segmentation. For calculation of description length, we will be working on designing more accurate formula. Due to our method is initial step of NLP task, in this experiment we use translation accuracy of building SMT system as evaluation of our method. However, we also suggest that our segmentation method could be evaluated with other machine translation system.

References

Shlomo Argamon, Navot Akiva, Amihood Amir, and Oren Kapah. 2004. Efficient unsupervised recursive word segmentation using minimum description length. In *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004)*, volume 2, pages 1058–1064, Genève, August.

Ruey-Cheng Chen. 2013. An improved MDL-based compression algorithm for unsupervised word segmentation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, volume 2, pages 166–170, Sofia, Bulgaria, August. Association for Computational Linguistics.

Carl De Marcken. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, Massachusetts Institute of Technology.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Jesús González-Rubio and Francisco Casacuberta. 2015. Improving the minimum description length inference of phrase-based translation models. In *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pages 219–227. Springer.

Peter Grünwald. 2005. A tutorial introduction to the minimum description length principle. In *Advances in Minimum Description Length: Theory and Applications*. MIT Press.

Daniel Hewlett and Paul Cohen. 2011. Fully unsupervised word segmentation with bve and mdl. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 540–545. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL-2007 on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X (MT summit X)*, volume 5, pages 79–86.

Adrien Lardilleux, François Yvon, and Yves Lepage. 2012. Hierarchical sub-sentential alignment with anymalign. In *Proceedings of the 16th annual conference of the European Association for Machine Translation (EAMT 2012)*, pages 279–286.

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*, pages 311–318. Association for Computational Linguistics.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Markus Saers, Karteek Addanki, and Dekai Wu. 2013. Unsupervised transduction grammar induction via minimum description length. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation (HyTra)*, pages 67–73.

Sami Virpioja, Jaakko J Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of Machine Translation Summit XI (MT Summit XI)*, pages 491–498.

Hua Yu. 2000. Unsupervised word induction using mdl criterion. In *Proceedings of the International Symposium of Chinese Spoken Language Processing (ISCSL 2000)*, Beijing.

Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2010. An efficient algorithm for unsupervised word segmentation with branching entropy and mdl. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 832—842. Association for Computational Linguistics, Oct.