Unwritten Languages Demand Attention Too! Word Discovery with Encoder-Decoder Models
Marcely Zanon Boito, Alexandre Bérard, Aline Villavicencio, Laurent Besacier

To cite this version:
Marcely Zanon Boito, Alexandre Bérard, Aline Villavicencio, Laurent Besacier. Unwritten Languages Demand Attention Too! Word Discovery with Encoder-Decoder Models. IEEE Automatic Speech Recognition and Understanding (ASRU), Dec 2017, Okinawa, Japan. hal-01592091

HAL Id: hal-01592091
https://hal.archives-ouvertes.fr/hal-01592091
Submitted on 22 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
UNWRITTEN LANGUAGES DEMAND ATTENTION TOO! WORD DISCOVERY WITH ENCODER-DECODER MODELS

Marcely Zanon Boito¹,², Alexandre Bérard¹, Aline Villavicencio² and Laurent Besacier¹

¹Laboratoire d’Informatique de Grenoble, Univ. Grenoble Alpes (UGA), France
²Institute of Informatics, UFRGS, Brazil

ABSTRACT

Word discovery is the task of extracting words from unsegmented text. In this paper we examine to what extent neural networks can be applied to this task in a realistic unwritten language scenario, where only small corpora and limited annotations are available. We investigate two scenarios: one with no supervision and another with limited supervision with access to the most frequent words. Obtained results show that it is possible to retrieve at least 27% of the gold standard vocabulary by training an encoder-decoder neural machine translation system with only 5,157 sentences. This result is close to those obtained with a task-specific Bayesian nonparametric model. Moreover, our approach has the advantage of generating translation alignments, which could be used to create a bilingual lexicon. As a future perspective, this approach is also well suited to work directly from speech.

Index Terms—Word Discovery, Computational Language Documentation, Neural Machine Translation, Attention models

1. INTRODUCTION

Computational Language Documentation (CLD) aims at creating tools and methodologies to help automate the extraction of lexical, morphological and syntactic information in languages of interest. This paper focuses on languages (most of them endangered and unwritten) spoken in small communities all across the globe. Specialists believe that more than 50% of them will become extinct by the year 2100 [1], and manually documenting all these languages is not feasible. Initiatives for helping with this issue include organizing tasks [2, 3] and proposing pipelines for automatic information extraction from speech signals [4, 5, 6, 7, 8].

Methodologies for CLD should consider the nature of the collected data: endangered languages may lack a well-defined written form (they often are oral-tradition languages). Therefore, in the absence of a standard written form, one alternative is to align collected speech to its translation in a well-documented language. Due to the challenge of finding bilingual speakers to help in this documentation process, the collected corpora usually are of small size.

One of the tasks involved in the documentation process is word segmentation. It consists of, given an unsegmented input, finding the boundaries between word-like units. This input can be a sequence of characters or phonemes, or even raw speech. Such a system can be very useful to linguists, helping them start the transcription and documentation process. For instance, a linguist can use the output of such a system as an initial vocabulary, and then manually validate the generated words. Popular solutions for this task are Nonparametric Bayesian models [9, 10, 11, 12, 13] and, more recently, Neural Networks [5, 8, 14]. The latter have also been used for related tasks such as speech translation [15, 16] or unsupervised phoneme discovery [17].

Contribution. This paper is the first attempt to leverage attentional encoder-decoder models for language documentation of a truly unwritten language. We show that it is possible, from very little data, to perform unsupervised word discovery with a performance (F-score) only slightly lower than that of Nonparametric Bayesian models, known to perform very well on this task in limited data settings. Moreover, our approach aligns symbols in the unknown language with words from a known language which, as a by-product, bootstraps a bilingual dictionary. Therefore, in the remainder of this paper, we will use the term word discovery (instead of word segmentation), since our approach does not only find word boundaries but also aligns word segments to their translation in another language.

Another reason why we are interested in attentional encoder-decoder models, is that they can easily be modified to work directly from the speech signal, which is our ultimate goal.

Approach. In a nutshell, we train an attention-based Neural Machine Translation (NMT) model, and extract the soft-alignment probability matrices generated by the attention mechanism. These alignments are then post-processed to segment a sequence of symbols (or speech features) in an unknown language (Mboshi) into words. We explore three improvements for our neural-based approach: alignment smoothing presented in [16], vocabulary reduction discussed in [18], and Moses-like symmetrization of our soft-alignment probability matrices. We also propose to reverse the translation direction, translating from known language words to
unknown language tokens. Lastly, we also study a semi-supervised scenario, where prior knowledge is available, by providing the 100 most frequent words to the system.

Outline. This paper is organized as follows: we present related work in Section 2, and the neural architecture, corpus, and our complete approach in Section 3. Experiments and their results are presented in Section 4 and 5, and are followed by an analysis in Section 6. We conclude our work with a discussion about possible future extensions in Section 7.

2. RELATED WORK

Nonparametric Bayesian Models (NB models) [19, 20] are statistical approaches that can be used for word segmentation and morphological analysis. Recent variants of these models are able to work directly with raw speech [10], or with sentence-aligned translations [12]. The major advantage of NB models for CLD is their robustness to small training sets. Recently, [18] achieved their best results on a subset (1200 sentences) of the same corpus we use in this work by using a NB model. Using the dpseg system\(^1\) [9], they retrieved 23.1% of the total vocabulary (type recall), achieving a type F-score of 30.48%.

Although NB models are well-established in the area of unsupervised word discovery, we wish to explore what neural-based approaches could add to the field. In particular, attention-based encoder-decoder approaches have been very successful in Machine Translation [21], and have shown promising results in End-to-End Speech Translation [15, 22] (translation from raw speech, without any intermediate transcription). This latter approach is especially interesting for language documentation, which often uses corpora made of audio recordings aligned with their translation in another language (no transcript in the source language).

While attention probability matrices offer accurate information about word soft-alignments in NMT systems [21, 15], we investigate whether this is reproducible in scenarios with limited amounts of training data. That is because a notable drawback of neural-based models is their need of large amounts of training data [23].

We are aware of only one other work using an NMT system for unsupervised word discovery in a low-resource scenario. This work [16] used an 18,300 Spanish-English parallel corpus to emulate an endangered language corpus. Their approach for unsupervised word discovery is the most similar to ours. However, we go one step further: we apply such a technique to a real language documentation scenario. We work with only five thousand sentences in an unwritten African language (Mboshi), as we believe that this is more representative of what linguists may encounter when documenting languages.

<table>
<thead>
<tr>
<th># types</th>
<th>#tokens</th>
<th>avg # tokens per sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mboshi Dev</td>
<td>1,324</td>
<td>3,133</td>
</tr>
<tr>
<td>Mboshi Train</td>
<td>6,245</td>
<td>27,579</td>
</tr>
<tr>
<td>French Dev</td>
<td>1,343</td>
<td>4,321</td>
</tr>
<tr>
<td>French Train</td>
<td>4,903</td>
<td>38,226</td>
</tr>
</tbody>
</table>

Table 1: Organization of the corpus in development (Dev, 514 sentences) and training (Train, 4,643 sentences) sets for the neural model.

3. METHODOLOGY

3.1. Mboshi-French Parallel Corpus

We use a 5,157 sentence parallel corpus in Mboshi (Bantu C25), an unwritten\(^2\) African language, aligned to French translations at the sentence level. Mboshi is a language spoken in Congo-Brazzaville, and it has 32 different phonemes (25 consonants and 7 vowels) and two tones (high and low). The corpus was recorded using the LIG-AIKUMA tool [24] in the scope of the BULB project [25].

For each sentence, we have a non-standard grapheme transcription (the gold standard for segmentation), an unsegmented version of this transcription, a translation in French, a lemmatization\(^3\) of this translation, and an audio file. It is important to mention that in this work, we use Mboshi unsegmented non-standard grapheme form (close to language phonology) as a source while direct use of speech signal is left for future work.

We split the corpus into training and development sets, using 10% for the latter. Table 1 gives a summary of the types (unique words) and tokens (total word counts) on each side of the parallel corpus.

3.2. Neural Architecture

We use the LIG-CRISTAL NMT system\(^4\), using unsegmented text input for training. The model is easily extendable to work directly with speech [15]. Our NMT models follow [21]. A bidirectional encoder reads the input sequence \(x_1, \ldots, x_A\) and produces a sequence of encoder states \(h_1, \ldots, h_A \in \mathbb{R}^{2 \times n}\), where \(n\) is the chosen encoder cell size. A decoder uses its current state \(s_t\) and an attention mechanism to generate the next output symbol \(z_t\). At each time step \(t\), the decoder computes a probability distribution over the target vocabulary. Then, it generates the symbol \(z_t\) whose probability is the highest (it stops once it has generated a special end-of-sentence symbol). The decoder then updates its state \(s_t\) with the generated token \(z_t\). In our task, since reference transla-

---

\(^1\)Available at http://homepages.inf.ed.ac.uk/sgwater/resources.html.

\(^2\)Even though it is unwritten, linguists provided a non-standard grapheme form, considered to be close to the language phonology.

\(^3\)For tokenization and lemmatization we used TreeTagger [26].

\(^4\)Available at https://github.com/eske/seq2seq.
tions are always available (even at test time), we always force feed previous ground-truth symbol \( w_t \) instead of the generated symbol \( z_t \) (teacher forcing).

\[
c_t = \text{attn}(h, s_{t-1}) \tag{1}
\]

\[
y_t = \text{output}(s_{t-1} \oplus E(w_{t-1}) \oplus c_t) \tag{2}
\]

\[
z_t = \arg \max y_t \tag{3}
\]

\[
s_t = \text{LSTM}(s_{t-1}, E(w_t) \oplus c_t) \tag{4}
\]

\( \oplus \) is the concatenation operator. \( s_0 \) is initialized with the last state of the encoder (after a non-linear transformation), \( z_0 = \langle \text{BOS} \rangle \) (special token), and \( E \in \mathbb{R}^{|V| \times n} \) is the target embedding matrix. The output function uses a maxout layer, followed by a linear projection to the vocabulary size \( |V| \).

The attention function is defined as follows:

\[
c_t = \text{attn}(h, s_t) = \sum_{i=1}^{A} \alpha_i^t h_i \tag{5}
\]

\[
\alpha_i^t = \text{softmax}(e_i^t) \tag{6}
\]

\[
e_i^t = v^T \tanh(W_1 h_i + W_2 s_t + b_2) \tag{7}
\]

where \( v, W_1, W_2, \) and \( b_2 \) are learned jointly with the other parameters of the model. At each time step \( t \) a score \( e_i^t \) is computed for each encoder state \( h_i \), using the current decoder state \( s_t \). These scores are then normalized using a softmax function, thus giving a probability distribution over the input sequence \( \sum_{i=1}^{A} \alpha_i^t = 1 \) and \( \forall i, 0 \leq \alpha_i^t \leq 1 \). The context vector \( c_t \) used by the decoder, is a weighted sum of the encoder states. This can be understood as a summary of the useful information in the input sequence for the generation of the next output symbol \( z_t \). The weights \( \alpha_i^t \) can be seen as a soft-alignment between input \( x_i \) and output \( z_t \).

Our models are trained using the Adam algorithm, with a learning rate of 0.001 and batch size \( N \) of 32. We minimize a cross-entropy loss between the output probability distribution \( p_t = \text{softmax}(y_t) \) and reference translation \( w_i \):

\[
L = \frac{1}{N} \sum_{i=1}^{N} \log p_t(s_i = w_1, \ldots, w_T | x_i) \tag{8}
\]

\[
\text{loss}(w_1, \ldots, w_T | x_i) = - \sum_{t} \sum_{j} \log p_{tj} \times \mathbb{1}(y_t = V_j) \tag{9}
\]

\[
p_{tj} = \frac{e^{y_{tj}}}{\sum_{k} e^{y_{tk}}} \tag{10}
\]

3.3. Neural Word Discovery Approach

Our full word discovery pipeline is illustrated in Figure 1. We start by training an NMT system using the Mboshi-French parallel corpus, without the word boundaries on the Mboshi side. This is shown as step 1 in the figure.

We stop training once the training loss stops decreasing. At this point, we expect the alignment model to be the most accurate on the training data. Then we ask the model to force-decode the entire training set. We extract soft-alignment probability matrices computed by the attention model while decoding (step 2).

Finally, we post-process this soft-alignment information and infer a word segmentation (step 3). We first transform the soft-alignment into a hard-alignment, by aligning each source symbol \( x_i \) with target word \( y_t \) such that: \( t = \arg \max_i \alpha_i^t \).

Then we segment the input (Mboshi) sequence according to these hard-alignments: if two consecutive symbols are aligned with the same French word, they are considered to belong to the same Mboshi word.

4. UNSUPERVISED WORD DISCOVERY EXPERIMENTS

For the unsupervised word discovery experiments, we used the unsegmented transcription in Mboshi provided by linguists, aligned with French sentences. This Mboshi unsegmented transcription is made of 44 different symbols.

We experimented with the following variations:

1. Alignment Smoothing: to deal with source (phones or graphemes) vs. target (words) sequence length discrepancy, we need to encourage many-to-one alignments between Mboshi and French. These alignments are needed in order to cluster Mboshi symbols into word-units. For this purpose, we implemented the alignment smoothing proposed by [16]. The softmax function used by the attention mechanism (see eq. 6) takes an additional temperature parameter: \( \alpha_i^t = \exp(e_i^t / T) \) \( \sum_j \exp(e_j^t / T) \). A temperature \( T \) greater than one\(^5\) will result in a less sharp softmax, which boosts many-to-one alignments. In addition,

\[\text{We use } T = 10, \text{ like the original paper [16].}\]
the probabilities are smoothed by averaging each score with the scores of the two neighboring words: \( \alpha_i^t \leftarrow (\alpha_{i-1}^t + \alpha_i^t + \alpha_{i+1}^t)/3 \) (equivalent to a low-pass filtering on the soft-alignment probability matrix).

2. **Reverse Architecture**: in NMT, the soft-alignments are created by forcing the probabilities for each target word \( t \) to sum to one (i.e. \( \sum_i \alpha_i^t = 1 \)). However, there is no similar constraint for the source symbols, as discussed in [16]. Because we are more interested in the alignment than the translation itself, we propose to reverse the architecture. The reverse model translates from French words to Mboshi symbols. This prevents the attention model from ignoring some Mboshi symbols.

3. **Alignment Fusion**: statistical machine translation systems, such as the Moses [27], extract alignments in both directions (source-to-target and target-to-source) and then merge them, creating the final translation model. This alignment fusion is often called symmetrization. We investigate whether this Moses-like symmetrization improves our results by merging the soft-alignments probability matrices generated by our base (Mboshi-French) and reverse (French-Mboshi) models. We replace each probability \( \alpha_i^t \) by \( \frac{1}{3}(\alpha_i^t + \beta_i^t) \), where \( \beta_i^t \) is the probability for the same alignment \( i \leftrightarrow t \) in the reverse architecture.

4. **Target Language Vocabulary Reduction**: to reduce vocabulary size on the known language, we replace French words by their lemmas. The intuition is that, by simplifying the translation information, the model could more easily learn relations between the two languages. For the task of unsupervised word discovery, this technique was recently investigated by [18].

The base model (Mboshi to French) uses an embedding size and cell size of 12. The encoder stacks two bidirectional LSTM layers, and the decoder uses a single LSTM layer. The reverse model (French to Mboshi) uses an embedding size and cell size of 64, with a single layer bidirectional encoder and single layer decoder.

We present in Table 2 the unsupervised word discovery task results obtained with our base model, and with the reverse model, with and without alignment smoothing (items 1 and 2). We notice that the alignment smoothing technique presented by [16] improved the results, especially for types.

Moreover, we show that the proposed reverse model considerably improves type and token retrieval. This seems to confirm the hypothesis that reversing the alignment direction results in a better segmentation (because the attention model has to align each Mboshi symbol to French words with a total probability of 1). This may also be due to the fact that the reverse model reads words and outputs character-like symbols which is generally easier than reading sequences of characters [28]. Finally, we achieved our best result by using the reverse model with alignment smoothing (last row in Table 2).

We then used this latter model for testing alignment fusion and vocabulary reduction (items 3 and 4). For alignment fusion, we tested three configurations using matrices generated by the base and reverse models. We tested the fusion of the raw soft-alignment probability matrices (without alignment smoothing), the fusion of already smoothed matrices, as well as this latter fusion followed by a second step of smoothing. All these configurations lead to negative results: recall reduction between 3% and 5% for tokens and between 1% and 9% for types. We believe this happens because by averaging the reverse model’s alignments with the ones produced by the base model (which does not have the constraint of using all the symbols) we degrade the generated alignments, more than exploiting information discovered in both directions.

Lastly, when running the reverse architecture (with alignment smoothing) using French lemmas (vocabulary reduction), we also noticed a reduction in performance. The lemmatized model version had a recall drop of approximately 2% for all tokens and types metrics. We believe this result could be due to the nature of the Mboshi language, and not necessarily a generalizable result. Mboshi has a rich morphology, creating a different word for each verb tense, which includes radical and all tense information. Therefore, by removing this from the French translations, we may actually make the task harder, since the system is forced to learn to align different words in Mboshi to the same word in French.

### Table 2: Unsupervised Word Discovery results with 4,643 sentences.

<table>
<thead>
<tr>
<th></th>
<th>TOKENS</th>
<th></th>
<th>TYPES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>F-score</td>
<td>Recall</td>
</tr>
<tr>
<td>Base Model (Mb-Fr)</td>
<td>7.16</td>
<td>4.50</td>
<td>5.53</td>
<td>12.85</td>
</tr>
<tr>
<td>Base Model (Mb-Fr) with Alignment Smoothing</td>
<td>6.82</td>
<td>5.85</td>
<td>6.30</td>
<td>15.00</td>
</tr>
<tr>
<td>Reverse Model (Fr-Mb)</td>
<td>20.04</td>
<td>10.02</td>
<td>13.36</td>
<td>18.62</td>
</tr>
<tr>
<td>Reverse Model (Fr-Mb) with Alignment Smoothing</td>
<td>21.44</td>
<td>16.49</td>
<td>18.64</td>
<td>27.23</td>
</tr>
</tbody>
</table>

[27]: 
[16]: 
[28]: 
[18]:
5. SEMI-SUPERVISED WORD DISCOVERY EXPERIMENTS

A language documentation task is rarely totally unsupervised, since linguists usually immerse themselves in the community when documenting its language. In this section, we explore a semi-supervised approach for word segmentation, using our best reverse model from Section 4.

To emulate prior knowledge, we select the 100 most frequent words in the gold standard for Mboshi segmentation. We consider this amount reasonable for representing the information a linguist could acquire after a few days. Our intuition is that providing the segmentation for these words could help improve the performance of the system for the rest of the vocabulary.

To incorporate this prior information to our system, we simply add known tokens on the Mboshi side of the corpus, keeping the remaining symbols unsegmented. This creates a mixed representation, in which the Mboshi input has at the same time unsegmented symbols and segmented words. Since languages follow Zipfian distributions [29] and we are giving to the model the most frequent words in the corpus, analysis is not done in terms of tokens, since this would be over-optimistic and bias the model evaluation, but only in terms of types. Results are presented in Table 3.

For types, we observed an increase of 2.4% in recall. This is not a huge improvement, considering that we are giving 100 words to the model. We discovered that our unsupervised model was already able to discover 97 of these 100 frequent words, which could justify the small performance difference between the models. In addition to the 100 types already known, the semi-supervised model found 50 new types that the unsupervised system was unable to discover.

Finally, it is interesting to notice that, while the performance increase is not huge, the semi-supervised system reduced considerably the number of types generated, from 11,266 to 7,473. This suggests that this additional information helped the model to create a better vocabulary representation, closer to the gold standard vocabulary.

<table>
<thead>
<tr>
<th></th>
<th>Unsupervised</th>
<th>Semi-supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>27.23</td>
<td>29.49</td>
</tr>
<tr>
<td>Precision</td>
<td>15.02</td>
<td>24.64</td>
</tr>
<tr>
<td>F-score</td>
<td>19.36</td>
<td>26.85</td>
</tr>
<tr>
<td># correct types</td>
<td>1,692</td>
<td>1,842</td>
</tr>
<tr>
<td># generated types</td>
<td>11,266</td>
<td>7,473</td>
</tr>
</tbody>
</table>

**Table 3:** Types results for the semi-supervised word discovery task (100 known words, 4,653 sentences).

<table>
<thead>
<tr>
<th>Reverse Model (Fr-Mb) with AS</th>
<th>Recall</th>
<th>Precision</th>
<th>F-score</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>dpseg</td>
<td>13.94</td>
<td>38.32</td>
<td>20.45</td>
<td>0.272</td>
</tr>
</tbody>
</table>

**Table 4:** Comparison between the NB model (dpseg) and the reverse model with alignment smoothing (AS) for unsupervised word discovery. The scores were obtained by averaging over three instances of each model.

### Fig. 2: Word frequency distribution of the three models and the gold standard distribution.

6. ANALYSIS

6.1. Baseline Comparison

As a baseline, we used dpseg [30, 31] which implements a Nonparametric Bayesian approach, where (pseudo)-words are generated by a bigram model over a non-finite inventory, through the use of a Dirichlet-Process.

We used the same hyper-parameters as [18], which were tuned on a larger English corpus and then successfully applied to the segmentation of Mboshi. We use a random initialization and 19,600 sampling iterations.

Table 4 shows our results for types compared to the NB model. Although the former is able to retrieve more from the vocabulary, the latter has higher precision, and both are close in terms of F-score. Additionally, ours has the advantage of providing clues for translation.

It is interesting to notice that our neural approach, which is not specialized for this task (the soft-alignment scores are only a by-product of translation), was able to achieve close performance to the dpseg method, which is known to be very good in low-resource scenarios. This highlights the potential of our approach for language documentation.

6.2. Vocabulary Analysis

To understand the segmentation behavior of our approach, we looked at the generated vocabulary. We compare our unsupervised and semi-supervised methods with the gold standard and the NB baseline, dpseg. The first characteristic
we looked at was the word distribution of the generated vocabularies. While we already knew that dpseg constrains the generated vocabulary to follow a power law, we observed that our approaches also display such a behavior. They produce curves that are as close to the real language distribution as dpseg (see Figure 2).

We also measured the average word length to identify under-segmentation and over-segmentation. To be able to compare vocabularies of varying sizes, we normalized the frequencies by the total number of generated types. The curves are shown in Figure 3. Reading the legend from left to right, the vocabulary sizes are 6,245, 2,285, 11,266, and 7,473.

Our semi-supervised configuration is the closest to the real vocabulary in terms of vocabulary size, with only 1,228 more types. All the approaches (including dpseg) over-segment the input in a similar way, creating vocabularies with average word length of four (Figure 3).

Since both dpseg and neural-based approaches suffer from the same over-segmentation problem, we believe that this is a consequence of the corpus used for training, and not necessarily a general characteristic of our approach in low-resource scenarios. For our neural approaches, another justification is the corpus being small, and the average tokens per sentence being higher at the French side (shown in Table 1), which can potentially disperse the alignments over the possible translations, creating multiple boundaries.

Moreover, as Mboshi is an agglutinative language, there were several cases in which we had a good alignment but wrong segmentation. An example is shown in Figure 4, where we see that the word “îmokwìwì” was split in two words in order to keep its alignment to both parts of its French translation “suis blessé”. This is also the case of the last word in this figure: Mboshi does not require articles preceding nouns, which caused misalignment. We believe that by exploiting translation alignment, we could constraint our segmentation procedure, creating a more accurate word discovery model. Finally, we were able to create a model of reasonable quality which gives segmentation and alignment information using only 5,157 sentences for training (low-resource scenario).

7. CONCLUSION

In this work, we presented a neural-based approach for performing word discovery in low-resource scenarios. We used an NMT system with global attention to retrieve soft-alignment probability matrices between source and target language, and we used this information to segment the language to be documented. A similar approach was presented in [16], but this work represents the first attempt at training a neural model with a real unwritten language based on a small corpus made of only 5,157 sentences.

By reversing the system’s input order and applying alignment smoothing, we were able to retrieve 27.23% of the vocabulary, which gave us an F-score close to the NB baseline, known for being robust to low-resource scenarios. Moreover, this approach has the advantage of naturally incorporating translation, which can be used for enhancing segmentation and creating a bilingual lexicon. The system is also easily extendable to work with speech, a requirement for most of the approaches in CLD.

Finally, as future work, our objective is to discover lexicon directly from speech, inspired by the encoder-decoder architectures presented in [15, 22]. We will also explore different training objective functions more correlated with segmentation quality, in addition to MT metrics. Lastly, we intend to investigate more sophisticated segmentation methods from the generated soft-alignment probability matrices, identifying the strongest alignments in the matrices, and using their segmentation as prior information to the system (iterative segmentation-alignment process).
8. REFERENCES


