Hybrid Language Segmentation for Historical Documents

Alfter, David and **Bizzoni, Yuri**
University of Gothenburg
{firstname.lastname}@gu.se

Abstract

English. Language segmentation, i.e. the division of a multilingual text into monolingual fragments has been addressed in the past, but its application to historical documents has been largely unexplored. We propose a method for language segmentation for multilingual historical documents. For documents that contain a mix of high- and low-resource languages, we leverage the high availability of high-resource language material and use unsupervised methods for the low-resource parts. We show that our method outperforms previous efforts in this field.

Italiano. *La segmentazione del linguaggio, la divisione di un testo multilingue in frammenti monolingue, è stata affrontata nel passato, ma la sua applicazione a documenti storici è rimasta in gran parte inesplorata. Proponiamo un metodo per la segmentazione linguistica di documenti storici multilingue. Per documenti che contengono sia lingue ad alta disponibilità di risorse che lingue sottorappresentate, utilizziamo a nostro vantaggio l'elevata disponibilità delle lingue con un'ampia gamma di risorse e impieghiamo sistemi non supervisionati per le parti che dispongono di un minor numero di risorse. Mostriamo che il nostro metodo supera gli sforzi precedenti in questo settore.*

1 Introduction

The computational processing of historical documents presents challenges that modern documents do not; often there is no standard orthography, and the documents may interleave multiple languages (Garrette et al., 2015). Furthermore, the languages used in the documents may by now be considered *dead* languages.

This work will address the issue of language segmentation, i.e. segmenting a multilingual text into monolingual fragments for further processing. While this task has been addressed in the past using supervised and weakly supervised methods such as trained language models (Řehůřek and Kolkus, 2009; King and Abney, 2013), unsupervised methods (Biemann and Teresniak, 2005; Yamaguchi and Tanaka-Ishii, 2012; Alfter, 2015a), the application to short messages (Porta, 2014; Alfter, 2015b) and the application to historical documents with regard to OCR tasks (Garrette et al., 2015), there is still room for improvement, especially concerning historical documents.

Due to the scarcity of multilingual corpora (Lui et al., 2014), a popular approach is to use monolingual training data. However, in the case of historical documents, the number of available texts in a given historical language might be too low to yield representative language models.

We propose a method that works on texts containing at least one high resource language and at least one low resource language. The intuition is to use supervised and weakly supervised methods for the high resource languages and unsupervised methods for the low resource languages to arrive at a better language segmentation; supervised methods derived from high-resource languages single out these languages while unsupervised algorithms tackle the remaining unknown language(s) and cluster them by similarity.

The presented approach is extendable to more than one high-resource language, in which case a separate language model has to be trained for each language; the approach is also applicable to more than one low-resource language, where the unsupervised methods are expected to produce an accurate split of all languages present.

2 Hybrid language segmentation

Let $D = w_1 ... w_n$ be a document consisting of the words w_1 to w_n. Let L_h be a character-level *n-gram* language model trained on data for a high resource language which occurs in the document D. We first apply the language model L_h to the document D and assign each word w_i the probability given by L_h (1).

$$\forall w_i \in D : P(w_i) = L_h(w_i) \tag{1}$$

The language model L_h is implemented as a trigram language model with non-linear back-off. For testing purposes, we trained a language model on a dump of the English Wikipedia (3 GB of compressed data).

Under the assumption that the text contains at least two languages with at least one word from each language, we determine the minimum probability P_{min} for a split (2). This probability corresponds to the lowest probability assigned by the language model L_h to any word in the text.

$$P_{min} = min_{i=1..n} P(w_i) \tag{2}$$

Next, we determine the maximum probability distance P_a between adjacent words (3) and the global maximum probability distance P_g between any two words (4).

$$P_a = max_{i=2..n}(\big|P(w_{i-1}) - P(w_i)\big|) \tag{3}$$

$$P_g = max_{i=1..n,j=1..n}(\big|P(w_i) - P(w_j)\big|) \tag{4}$$

We also calculate the mean probability P_{mean} between the two adjacent words which maximize P_a (5).

$$P_{mean} = \frac{P(w_i) + P(w_j)}{2} \tag{5}$$

Finally, we calculate the sharpest drop in probabilities and define $P_{mindrop}$ as the probability at the lowest point of the drop (6).

$$\begin{aligned} P_{mindrop} = & max_{i=3..n}(\big|P(w_{i-2}) - P(w_{i-1})\big| \\ & + \big|P(w_{i-1}) - P(w_i)\big|) \end{aligned} \tag{6}$$

We then set a preliminary language split threshold P_{split} based on P_{min}, P_a, P_g, P_{mean} and $P_{mindrop}$ (7).

$$P_{split} = \frac{\frac{\frac{P_a + P_g}{3} + P_{mean}}{2} + P_{mindrop}}{2} \tag{7}$$

In a first step, every word w_i with a probability P above the split threshold P_{split} is considered to belong to the high resource language modeled by L_h and is tagged as such, while every word w_j with a probability P below the split threshold is considered as belonging to an unknown language and is left untagged.

In a second step, all untagged words are clustered by similarity. This is done by using language model induction (Alfter, 2015a). All words left untagged by the previous step are regarded as one text. From the first word w_1, an initial language model L_i is created. The next word w_2 is tested against the initial model. If the probability $P(w_2|L_i)$ exceeds a certain threshold value, the model is updated with w_2, otherwise a new model is created. In this way, we iterate through the text, creating language models as necessary. The same procedure is done starting from the last word and moving towards the beginning of the text. From the two sets of language model inductions (forward, backward), the most similar models according to their n-gram distribution are then merged. This process is repeated, keeping the previously merged models, until no more models are induced.

Each word is then tagged with the language model L_m ($\approx$ cluster) which maximizes $P(w|L_m)$.

Finally, all words are evaluated in a local context using variable-length Markov Models (VMM). This step aims at eliminating inconsistencies, detecting other-language inclusions and merging back together same-language fragments. Řehůřek and Kolkus (2009) use a similar technique, but they use a fixed-width sliding window while we use a variable window size based on context.

For each word w_i, we look at its tag t_i. We then consider all the words immediately to the left of w_i and all the words immediately to the right of w_i that have a tag different from t_i. From these words, we create local context language models left (L_l) and right (L_r). We calculate the similarity between L_l and L_r as well as the similarity of w_i to L_l and L_r. There are different possible scenarios:

1. L_l is similar to L_r
   (a) w_i is similar to L_l or L_r
   (b) w_i is dissimilar to L_l or L_r
2. L_l is dissimilar to L_r
   (a) w_i is similar to L_l
   (b) w_i is similar to L_r
   (c) w_i is dissimilar to L_l and L_r

In case 1a, we assimilate the tag of w_i to the tag of either L_l or L_r; in that case, the labels for L_l and L_r are the same. In case 1b, w_i is probably an other-language inclusion, since it is dissimilar to its context, while the left and right contexts are similar. In case 2a, we assimilate the tag of w_i to the tag of L_l, and similarly in case 2b, we assimilate the tag of w_i to L_r. In case 2c, w_i is dissimilar to its context and the left and right contexts are also dissimilar. In this case, we leave the tag unchanged.

The following sections describe the data used for evaluation as well as the results.

3 Data and Evaluation

> Pacati, [Ved. pacati, Idg. *peqŭō, Av. pac-; Obulg. peka to fry, roast, Lith, kepū bake, Gr. pέssw cook, pέpwn ripe] to cook, boil, roast Vin. IV, 264; fig. torment in purgatory (trs. and intrs.): Niraye pacitvā after roasting in N.S.II, 225, PvA. 10, 14. – ppr. pacanto tormenting, Gen. pacato (+Caus. pācayato) D. I, 52 (expld at DA. I, 159, where read pacato for paccato, by pare daṇḍena pīḷentassa). – pp. pakka (q.v.). <->Caus. pacāpeti & pāceti (q. v.). – Pass. paccati to be roasted or tormented (q. v.). (Page 382)

In the absence of better comparable data, we re-use the Pali dictionary data entries presented in Alfter (2015a) and compare our calculated language segmentation to the segmentation presented in Alfter (2015a).

The extract shown corresponds to the fifth Pali text used in the experiments. It shows among others some of the languages used, the unclear boundaries between languages, abbreviations, symbols and references. Monolingual stretches tend to be short with interspersed language inclusions.

Based on the findings in Alfter (2015a) that neither a high Rand Index nor a high F-score alone yield good segmentations, but a combination of high Rand Index and F-score yield good segmentations, we have adopted a new measure of goodness-of-segmentation G_s, which is the arithmetic mean of the Rand Index and F5 score (8).

$$G_s = \frac{RI + F5}{2} \tag{8}$$

Due to how precision and recall are calculated in the context of cluster evaluation, setting $\beta > 1$, and thus placing more emphasis on recall, penalizes the algorithm for clustering together data points that are separated in the gold standard and lowers the impact splitting of data points which are clustered together in the gold standard. Indeed, it is preferable to have multiple clusters of a certain language than to have clusters of mixed languages. Thus, we use F5 ($\beta = 5$) instead of F1 scores.

We have found left context assimilation to be working better than right context assimilation or both side context assimilation. We therefore use only left context assimilation and leave out the other two options.

4 Results

The following table shows our results (Hybrid Language Segmentation, HLS) compared to the results given in Alfter (2015a) (Language Model Induction, LMI). We converted the scores given in Alfter (2015a) to the new compound score G_s. The baselines from Alfter (2015a) are also indicated. AIO indicates the baseline where each word is thrown into the same cluster; there is only one cluster (all-in-one). AID indicates the baseline where each word is separated into its own cluster; there is one cluster per word (all-in-different).

Text	AIO	AID	LMI	HLS
Pali 1	0.3174	0.4643	0.5296	0.6665
Pali 2	0.3635	0.5188	0.7662	0.5916
Pali 3	0.4996	0.3071	0.4700	0.6056
Pali 4	0.4047	n/a	n/a	0.4730
Pali 5	0.5848	0.2833	0.4402	0.5863

Table 1: Results

As can be seen from the results, our approach outperforms the baselines as well as the

purely unsupervised language model induction approach except for one data point where the language model induction produced an almost perfect clustering whereas the hybrid language segmentation method did not.

5 Discussion

A big problem with the dictionary data is that it is transcribed in a noisy manner. This is not immediately clear from looking at the data, but on closer inspection, it can be seen that some symbols like commas and full stops are rendered with non-standard Unicode characters (Unicode codepoint U+FF0C (FULLWIDTH COMMA) and Unicode codepoint U+FF0E (FULLWIDTH FULL STOP)) which break the chosen whitespace tokenization method. This results in chunks that are bigger than they should be, often containing multiple languages. We can also see that the transcription of Greek characters were rendered as character that look alike but are not actually Greek characters (see the quote at the beginning of section 3).

If we look more closely at the results, we can see that our approach tends to be overly confident when assigning words to the high-resource language, which in this case is English. This includes words that clearly are not English, such as '°itar' and '°ātar'[1]. The following example (Pali 1) shows the full dictionary entry.

> [n. ag. fr. abhijjhita in med. function] one who covets M <smallcaps>i.</smallcaps> 287 (T. abhijjhātar, v. l. °itar) = A <smallcaps>v.</smallcaps> 265 (T. °itar, v. l. °ātar).

The poor discriminatory power of the model is probably related to the training data. While the English Wikipedia offers a huge amount of training data, it also includes many non-English words in explanations and on pages about non-English non-translatable terms for example. Thus, the resulting language model is noisy.

It might be possible to increase accuracy by changing the split threshold P_{split}, but while choosing a higher P_{split} will effectively reduce the amount of erroneous English tags, it will also decrease the amount of correctly tagged words. It is possible that the unsupervised approach followed by the local context smoothing might re-assign the English words to the English model or at least to a consistent, second model. However, this remains to be tested. We think that simply using more 'pure' English training data will improve the language model's accuracy.

As for local context smoothing, we have not reached conclusive results. While in some cases, it succeeds in re-assigning the correct tag to a previously incorrectly tagged word, it also induces errors by erroneously re-tagging previously correct tags. This is most probably due to the short monolingual fragments in our data; longer monolingual fragments would yield more reliable language models. In connection to this, calculating similarity based on small contexts seems problematic. Another problem are non-words and their treatment. We have chosen not to cross non-word boundaries when calculating local context, but doing so might improve the results.

Finally, we have only tested the approach with one high resource language and a multitude of low-resource languages. It would be interesting to test the method more extensively using more high resource language models (which in turn might interfere with each other).

6 Conclusion

We have introduced a hybrid language segmentation method which leverages the presence of high-resource language content in mixed language historical documents and the availability of the necessary resources to build language models, coupled with an unsupervised language model induction approach which covers the low-resource parts. We have shown that our method outperforms the previously introduced unsupervised language model induction approach.

We have also found that our method seems to work both on longer texts and on shorter texts, whereas the approach described in Alfter (2015a) seems to be working better on shorter texts such as Twitter messages.

The local context approach yields inconclusive results. This is most probably due to the similarity measure used and the small size of the context. We would need, if possible, a better similarity measure for small language models or another method of evaluating the word in respect to its context.

[1] Here, ° stands for the root of the head word of the entry, so °itar should be read 'abhijjhitar' and °ātar should be read 'abhijjhātar'

References

Alfter, D. (2015a). Language Segmentation. Master's thesis, Universität Trier.

Alfter, D. (2015b). Language segmentation of twitter tweets using weakly supervised language model induction. *TweetMT @ SEPLN*.

Biemann, C. and Teresniak, S. (2005). Disentangling from babylonian confusion–unsupervised language identification. In *Computational Linguistics and Intelligent Text Processing*, pages 773–784. Springer.

Garrette, D., Alpert-Abrams, H., Berg-Kirkpatrick, T., and Klein, D. (2015). Unsupervised code-switching for multilingual historical document transcription. In *Proceedings of NAACL*.

King, B. and Abney, S. P. (2013). Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies*, pages 1110–1119.

Lui, M., Lau, J. H., and Baldwin, T. (2014). Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2:27–40.

Porta, J. (2014). Twitter Language Identification using Rational Kernels and its potential application to Sociolinguistics. *TweetLID @ SEPLN*.

Řehůřek, R. and Kolkus, M. (2009). Language identification on the web: Extending the dictionary method. In *Computational Linguistics and Intelligent Text Processing*, pages 357–368. Springer.

Yamaguchi, H. and Tanaka-Ishii, K. (2012). Text segmentation by language using minimum description length. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 969–978. Association for Computational Linguistics.