A hybrid pipeline of rules and machine learning to filter web-crawled parallel corpora

Eduard Barbu
Institute of Computer Science
University of Tartu
Tartu, Estonia
eduard.barbu@ut.ee

Verginica Barbu Mititelu
Research Institute for Artificial Intelligence
Romanian Academy
Bucharest, Romania
vergi@racai.ro

Abstract

A hybrid pipeline comprising rules and machine learning is used to filter a noisy web English-German parallel corpus for the Parallel Corpus Filtering task. The core of the pipeline is a module based on the logistic regression algorithm that returns the probability that a translation unit is accepted. The training set for the logistic regression is created by automatic annotation. The quality of the automatic annotation is estimated by manually labeling the training set.

1 Introduction

The task "Parallel Corpus Filtering" presented a noisy web crawled parallel corpora (English-German) whose English side contains one billion words. The participants had to select two "clean" subsets consisting of 10 million words and 100 million words, respectively. The quality of the two subsets was determined by the BLEU score of a statistical machine translation (based on Moses) and a neural machine translation system (Marian) trained on these subsets. The BLEU scores were computed for multiple not disclosed sets.

The parallel corpus filtering task bears similarity to translation memory cleaning task and Quality estimation task.

Some systems that spot false translation units in translation memories are surveyed in Barbu (2016). One of the most successful systems is trained not only on features related to translation quality, but also on features related to grammatical errors and features related to fluency and lexical choice (Wolff, 2016).

Given the similarity between the translation memory cleaning task and this task we have adapted part of our system for cleaning the translation memories. The system requires supervision and word alignment knowledge. However, the "Parallel Corpus Filtering" task specifications restrict the usage of external parallel corpora and allow minimum alignment information. Therefore, we had to re-engineer the above mentioned system and produce a pipeline that respects the task requirements.

In the next section we present the re-engineered pipeline. The section 3 shows an in-house evaluation and in the last section we draw the conclusions.

2 Pipeline description

The pipeline for finding the best translation units for the Parallel corpus filtering task is shown in figure 1. The pipeline consists of three modules, which we describe below.

The module **Filtering Rules** filters those translation units that are not good to train machine translation systems on because they are either too short or are prone to errors. The discarded units have less than 10 words in source or target, or the language codes assigned by the language detector Cybozu[1] do not coincide with the expected language codes ("en" for the source segment and "de" for the target segment), or have a Church-Gale score (Gale and Church, 1993) that is less than -4 or greater than 4. In the task submission the translation units that do not fulfill the above criteria have a score equal to -100. The initial number of translation units is 104.002.521. After filtering there remain 11.030.014 translation units. The English side of the remaining units contains 269.949.547 words corresponding to 241.984.520 German words. From this filtered set the two subsets required by the task description are selected.

The module **Machine Learning** is the core of the pipeline. Because the manual annotation was

[1] https://github.com/shuyo/language-detection

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 880–884

https://doi.org/10.18653/v1/W18-6475

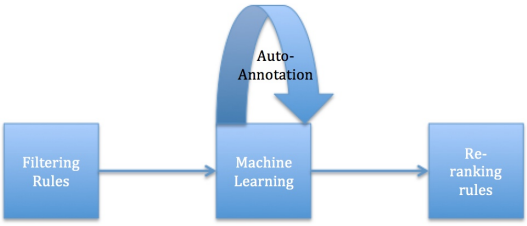

Figure 1: The pipeline for translation units selection

not allowed, the training set was generated by a simple heuristic rule. From the translation units not scored by the previous module we have drawn randomly approximately 1700 translation units. These units are annotated automatically in the following way. If the Hunalign (Varga et al., 2005) score, provided with the test file, for a translation unit is higher than a fixed threshold 0.9[2] we consider that translation unit as a positive example. If the Hunalign score is less or equal to the threshold the translation unit is a negative example. In section 3 we evaluate how accurate the automatic annotation is.

The **Machine Learning** module uses three kinds of features: *Presence/Absence* features, *Alignment Features* and *Fluency Features*. The feature values are all numerical because for classification we use scikit-learn machine learning toolkit (Pedregosa et al., 2011).

1. **Presence/Absence** features. This category of features signal the presence/absence of an entity in source or target segments. The features capture the intuition that if an entity is present in the source segment and if the target segment is a translation of the source segment it is very probable that the same entity is present in the target segment.

    - *Entity Features*. These features are *tag*, *URL*, *email*, *name entity*, *punctuation*, *number*, *capital letters*, *words in capital letters*. The value of these features is 1 if the source or target segments contain a tag, URL, email, name entity, punctuation, capital letters or words written in capital letters, otherwise is 0.
    - *Entity Similarity Features*. For features *tag*, *URL*, *email*, *name entity*, *punctuation*, *number*, the cosine similarity between the source and target segments entity features vectors is computed. If the respective features are present in the source segment and the target segment is the translation of the source we expect that the system learns the range of the admissible similarity values.
    - *Capital letters word difference*. The value of this feature is the ratio between the difference of the number of words containing at least a capital letter in the source segment and the target segment and the sum of the capital letter words in the translation unit. It is complementary to the feature *capital letters*.
    - *Only capital letters difference*. The value of the feature is the ratio between the difference of the number of words containing only capital letters in the source segment and the target segments and the sum of only the capital letter words in the translation unit. It is complementary to the feature *words in capital letters*.

2. **Alignment Features**. The idea behind alignment features is that sentence alignments, or the information that can help to decide if an alignment is likely or not, provide an important clue for the hypothesis that source and target segments are translations.

    - *language difference*. If the language codes identified by Cybozu language detector for the source and target segments coincide with the language codes declared for the same source and target segments, then the feature value is 1, otherwise the feature value is 0. As we have seen, the English and German segments have more than 10 words, therefore the language detector has enough information to return the segment language with good precision.
    - *Gale-Church score*. This feature is the slightly modified Gale-Church score described in the equation 1 and introduced in (2011). This score reflects the idea that the length of the source (l_s) and target segments (l_d) that are true translations is correlated. We expect that the classifiers learn the threshold that

[2]The threshold value comes from our previous experience with Hunalign aligner.

separates the positive and negative examples. However, relying exclusively on the Gale-Church score is tricky because there are cases when a high Gale-Church score is perfectly legitimate. For example, when the acronyms in the source language are expanded in the target language.

$$CG = \frac{l_s - l_d}{\sqrt{3.4(l_s + l_d)}} \quad (1)$$

- *Hunalign score*. This is the score returned by Hunalign sentence aligner and was provided by the task organizers. The score depends on the quality of the English-German dictionary used by the aligner.

3. **Fluency Features**. These features values correlate with fluency of the translation units in source and target languages.

   - *Perplexity* To capture the fluency of the source and target we compute the perplexity of the segments in English and German using KenLM toolkit (Heafield, 2011). The KenLM language model was trained as advised on the shared-task web page - on the WMT 2018 news translation task data for German-English from which we have eliminated the Paracrawl parallel corpus. Moreover, we have also run Cybozu language detector to eliminate sentences that are not identified as written in English or German. Thus, the English corpus for training KenLM language model has 5.802.775 sentences and 126.831.658 words and the German corpus has 5.673.375 and 116.360.460 words.

The classification algorithm used by the **Machine Learning** module is logistic regression. For each filtered translation unit this module outputs the probability score that the respective unit is positive. One hopes that this probability score correlates with the translation unit quality.

The last module, **Re-ranking rules**, comprises a set of rules to re-score the probability scores outputted by the previous module. It implements the following rules :

1. *Same Digits Rule*. This rule states that if the target segment is a translation of the source segment, and the source segment contains some digits, then the target segment should contain the same digits, possibly in a different order. If this is not the case, the translation unit is re-scored by subtracting 1 from its probability score. Please, notice that the rule allows for the dates to be written in different formats. For example, if the source segment contains the date "02/01/2001" (format mm/dd/yyyy) and in the target segment the date is written as "01/02/2001" (format dd/mm/yyyy), then the translation unit is not re-scored.

2. *Same Numbers*. This rule states that if the target segment is a translation of the source segment, and the source segment contains some numbers, then the target segment should contain the same numbers possibly in a different order. If a translation unit passes the first rule by chance and if it does not pass this rule, then it will be downgraded subtracting 1 from is probability score.

3. *Rule URL*. This rule applies to those translation units that contain Uniform Resource Locators like web addresses, for example. If the length of the web address is longer than the portion of normal text in the source or target segment, then the translation unit is re-scored by subtracting 1 from its probability score.

4. *Rule Tags*. If the source and target segments are translations and they contain tags, then we expect that the tags are the same. If this is not the case 1 is subtracted from the translation unit probability score.

Finally, to ensure diversity among the best rated translation units that comprise the first evaluated set containing 10 millions of words we compute the cosine similarity between the English segments. We keep in the first set only those translation units whose cosine similarity (computed between English segments) is less than 0.85[3]

[3]To compute the cosine matrix we have used "TfidfVectorizer" from "sklearn". Unfortunately, on our server we could not compute the matrix for all units and had to restrict to compute matrices with 30000 lines.

Confusion Matrix	Predicted 1	Predicted 0
Actual 1	1010	233
Actual 0	67	404

Table 1: The Confusion Matrix

3 Evaluation

We have manually annotated the automatically annotated pairs used to train the logistic regression algorithm. A non-native German language speaker has annotated this set with the label "1" if the translation unit is accurate and "0" otherwise. Two examples of annotated translation units are given bellow.

- **A correctly automatic annotated translation unit**
  - In a nutshell: the usage of the machinery for sifting, to loosen and rasp, or to prepare powdery substances and hygroscopic materials.
  - Kurz: Überall zum maschinellen Passieren, Auflockern und Raspeln oder zum Aufbereiten pulverförmiger Massen und hygroskopischer Materialien.
- **An incorrectly automatic annotated translation unit**
  - Large swimming pool and gym, for those who want to combine open air and relaxing activities with indoor training
  - Die Räume liegen direkt neben dem großen Pool und dem Fitnessraum, für all diejenigen die zu den vielzähligen Outdoor-Aktivitäten ein Trainigsprogramm in den Innenräumen kombinieren möchten.

In both examples the Hunalign score is higher than the fixed threshold but only the first example is correctly annotated automatically. The automatic annotator is a binary classifier and we can evaluate this classifier as is customary by comparing its annotation with a gold standard (the manual annotation). As one can see from the confusion matrix in table 1 the training set is imbalanced with only 27 percent negative examples.

The precision, recall, F1-score and the balanced accuracy for the positive and negative classes are shown in table 2. All scores are high, showing

Measure	Value
Precision Positive class	0.93
Precision Negative class	0.63
Recall Positive class	0.81
Recall Negative class	0.86
F1 score Positive class	0.87
F1 score Negative class	0.73
Balanced Accuracy	0.83

Table 2: Classification results

that the heuristic based on Hunalign threshold is a good one. However, one should also consider that the automatically annotated set is not a representative sample of the test set provided by the organizers of the task. To have a representative sample much more translation units should have been annotated.

The annotation errors are mitigated by the fact that the Logistic regression classifier trained on the automatically annotated set will return the probability of the positive class. If the probability correlates with translation unit quality, then some translation units, even if not perfect, could be useful for training machine translation systems.

We counted some cases when the sentence in one language translates the sentence in the other language, but, at the same time, is more informative, as it contains another part for which there is no translation in the other language. Another worth making remark is the existence of many Bible passages, at least in the set we have manually annotated. They have lexical, morphological and syntactic characteristics which are specific to this kind of writing and which, when applied to other kinds of writing, will give inappropriate results. Although accepted as useful for MT in this task, they are probably good only for translating similar kinds of texts (i.e., religious ones).

A much better evaluation is provided by the task organizers. They have determined the quality of the cleaning performed by the teams by the BLEU score of a statistical machine translation (based on Moses) and a neural machine translation system (Marian) trained on two subsets as explained in the introduction section. There were 48 submissions and our system ranked in the range 22 - 31 depending on the subset and machine translation system used in the evaluation. For details regarding shared task preparation, the official results table and a survey of the methods used by the partic-

ipating systems one should consult (Koehn et al., 2018).

4 Conclusions

In this paper we have presented a hybrid pipeline comprising rules and machine learning that was used to filter a noisy web English-German parallel corpus. The core of the pipeline is a logistic regression algorithm trained on an automatic annotated set. We have seen that the heuristic used to automatically annotate the training set is very good having 0.83 balanced accuracy (computed against the same set manually annotated). The pipeline also contains rules for re-scoring the translation units and a module based on cosine similarity to enhance the diversity of translation unit selection.

The core system, the manually annotated set and the python script for the evaluation procedure described in section 3 are publicly available on github[4].

Acknowledgments

This study was supported by the Estonian Ministry of Education and Research (IUT20-56).

References

Eduard Barbu, Carla Parra Escartín, Luisa Bentivogli, Matteo Negri, Marco Turchi, Constantin Orasan, and Marcello Federico. 2016. The first automatic translation memory cleaning shared task. *Machine Translation*, 30(3):145–166.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *COMPUTATIONAL LINGUISTICS*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Jörg Tiedemann. 2011. *Bitext Alignment*. Number 14 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael, CA, USA.

D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy. 2005. Parallel corpora for medium density languages. In *Recent Advances in Natural Language Processing (RANLP 2005)*, pages 590–596.

Friedel Wolff. 2016. Combining off-the-shelf components to clean a translation memory. *Machine Translation*, 30(3):167–181.

[4]https://github.com/SoimulPatriei/LogisticRegression-Shared-Task-Parallel-Corpus-Filtering.git