Building Parallel Corpora Using Cross-Lingual Bag-Of-Words

Taycir Yahmed

July 2018

Mining parallel corpora

Training machine translation models requires a huge amount of parallel data. Consequently, there has been many works suggesting different methods to build bilingual corpora, leading to the construction of reliable training datasets for machine translation systems.

However, the problem is still prominent for the below use-cases:

- Low-resource setup: Although for some language pairs, we have parallel datasets with a *convenient* size (e.g. around 50 millions sentences for French - English), this is not the case for all language pairs. Indeed, low-resource languages [1] do not have as much parallel data making it hard to train reliable translation models to and from these languages.
- Specialization setup: Furthermore, machine translation is sensitive to context. Thus, any available specialized data can have a strong influence on the model's performance for a specific domain. For instance, using medical data when training the model enhances its performance on prescriptions' translation.

  Note that there are various domain control strategies for machine translation, such as adding the domain tag as an additional feature or adding a special token to the sentence when training and translating; this is not, however, the core of this article. [1]

Due to the aforementioned reasons, there is still room for designing and implementing solutions for building parallel corpora. In the following sections, I present a solution for matching multilingual documents in order to construct a parallel corpus.

[1] Low-resource languages can be identified as languages that do not have enough software, data, tools or resources available

CLBOW: Cross-Lingual Bag-Of-Words

When designing an algorithm to match cross-lingual documents, the first reflex is to represent all available documents in numerical vectors. However, to compare these documents, the vectorial representations should be language-independent or *cross-lingual*, meaning that semantically similar documents should be close in the multidimensional representation space.

Although most recent research works focus on multilingual word embeddings as a numerical representation of text data [2], here we present a generalization of Bag-Of-Words to a cross-lingual setup, where we represent all documents in the same space irrespectively of their language. Below is the explicit implementation of the algorithm:

Result: Cross-lingual numerical representation of a document

Input: a document in a random language + a multilingual dictionary.
Initialization: vector ⟵ numerical representation of the document (zeros);

for *entry in cross-lingual dictionary* **do**
  for *language in languages* **do**
    if *entry.language in document* **then**
      vector.entry ⟵ vector.entry + 1;
    end
  end
end

Algorithm 1: Implementation of CLBOW: Cross-Lingual Bag of Words

	livre book	école school	réunion meeting	... ...	amélioration improvement
	1	0	1	...	1
Doc A	**1**	**1**	**0**	...	**0**
	...	...	...	...	...
Doc B	**1**	**0**	**1**	...	**1**

Figure 1: Illustration of CLBOW: Cross-Lingual Bag-Of-Words

Notes:

- In the above illustration, the decoding of only two languages is presented for simplicity purposes; nevertheless the suggested implementation is ex-

tended to many languages.

- Furthermore, it can handle polysemy since at each decoding step t, not only one translation of the word w_t is considered but its different translations.
- This version of BOW can provide both binary and numerical representation of the documents. By numerical, I refer to the extension of TF-IDF (Term Frequency - Inverse Document Frequency) to a cross-lingual setup.

Application to parallel corpora construction

Thanks to the previous algorithm, cross-lingual vectorial representations of the documents are calculated. Afterwards, a search for the closest document of a different language is performed using the minimization of the cosine distance and with regards to a threshold corresponding to the typical length ratio for the language pair. For instance, this threshold is equal to 1.5 for French-English bilingual corpora. A maximum accepted distance between a document and a candidate translated version is also considered, to discriminate documents having the same template (headers, footers, etc.). In my various experiments, this threshold is equal to 0.6.

Using this method, classes of equivalence representing each the multilingual versions of the same document are retrieved. For example, a class of equivalence can be represented as the following:

```
{'fr': 'Regle FR 29-01-2018.pdf',
'en': 'CS1548325.pdf',
'pt' : 'Regra 29-01-2018.pdf'
}

{'fr': 'Doc 12052005.pdf',
'en': 'To print.pdf',
'de' : 'pr12052005.pdf'
}
```

Figure 2: Examples of the resulting classes of equivalence

Below is the detailed algorithm:

Result: Classes of equivalence each representing the multilingual versions of the same document.

Input: a pool of non-matched multilingual documents.
Initialization:
matrix $\longleftarrow$ numerical representation of the documents;
selected_matches $\longleftarrow$ empty list;

for *document in documents* **do**
  matrix.document.vector $\longleftarrow$ cross-lingual BOW representation.
  matrix.document.lang $\longleftarrow$ detected language.
  matrix.document.length $\longleftarrow$ number of words.
end

for doc_a *in documents* **do**
  matches $\longleftarrow$ empty dict;
  for doc_b *in documents* **do**
    $relative\_length \longleftarrow \frac{max(doc_b.length, doc_a.length)}{min(doc_b.length, doc_a.length)}$
    if $doc_a.lang \neq doc_b.lang$ ***and*** $relative\_length \leq threshold_{ab}$ **then**
      $dis\_ab \longleftarrow$ cosine distance ($doc_b.vector$, $doc_a.vector$)
      if $dis\_ab \leq threshold\_dis$ **then**
        if $doc_b.lang \in matches$ **then**
          if $matches.doc_b.lang.dis < dis\_ab$ **then**
            matches.$doc_b.lang$.dis $\longleftarrow$ dis_ab
            matches.$doc_b.lang$.pred $\longleftarrow$ doc_b
          end
        else
          matches.$doc_b.lang$ $\longleftarrow$ empty dict;
          matches.$doc_b.lang$.dis $\longleftarrow$ dis_ab
          matches.$doc_b.lang$.pred $\longleftarrow$ doc_b
        end
      end
    end
  end
  selected_matches $\longleftarrow$ *selected_matches* + matches
end

Algorithm 2: Multilingual document matching

To build bilingual corpora, I consider sequentially pairs of languages. Then on each pair of documents, I apply sentence alignment using the algorithm BLEUAlign [3]. This will provide a bilingual parallel corpus for each data source relevant to a specific domain. These corpora are then used to train and specialize machine translation systems and using them enabled a good enhancement in BLEU score [4]. Generally, if $\Delta BLEU$ is the difference between the BLEU on a standard dataset and a specialized dataset of the general model, you should

expect to gain around $\Delta BLEU$ on the specialized dataset using the augmented model.

Conclusion

The here-presented pipeline enabled the construction of a specialized bilingual corpus, that I used to enhance the performance of translation models both on standard datasets and on specialized data (financial, medical, etc.). Other improvements are however to be tested in the near future, including neural encoding of multilingual documents.

References

[1] Catherine Kobus, Josep Maria Crego, and Jean Senellart. Domain control for neural machine translation. *CoRR*, abs/1612.06140, 2016.

[2] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[3] Rico Sennrich and Martin Volk. MT-based Sentence Alignment for OCR-generated Parallel Texts. Riga, Latvia, 2010.

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.