Proceedings of the 13th Workshop on Building and Using Comparable Corpora, pages 44–48
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Reducing the Search Space for Parallel Sentences in Comparable Corpora

Rémi Cardon, Natalia Grabar
CNRS, Univ. Lille, UMR 8163 STL - Savoirs Textes Langage
F-59000 Lille, France
{remi.cardon, natalia.grabar}@univ-lille.fr

Abstract

This paper describes and evaluates three methods for reducing the research space for parallel sentences in monolingual comparable corpora. Basically, when searching for parallel sentences between two comparable documents, all the possible sentence pairs between the documents have to be considered, which introduces a great degree of imbalance between parallel pairs and non-parallel pairs. This is a problem because, even with a highly performing algorithm, a lot of noise will be present in the extracted results, thus introducing a need for an extensive and costly manual check phase. We propose to study how we can drastically reduce the number of sentence pairs that have to be fed to a classifier so that the results can be manually handled. We work on a manually annotated subset obtained from a French comparable corpus.

Keywords: Parallel corpus creation, syntax, French

1. Introduction

Monolingual parallel corpora are useful for a variety of sequence-to-sequence tasks in natural language processing, such as text simplification (Xu et al., 2015), paraphrase acquisition (Deléger and Zweigenbaum, 2009) or style transfer (Jhamtani et al., 2017).

In order to build such parallel corpora, the typical approach is to start from comparable corpora and extract sentence pairs that share the same meaning. For instance, the participants of the BUCC 2017 shared task had to address this problem using bilingual corpora (Zweigenbaum et al., 2017). One major obstacle is that, when considering two documents *A* and *B*, every single sentence from *A* has to be evaluated against every single sentence of *B*, when document metadata cannot be used to make assumptions as to where to look for corresponding sentences. This produces a large amount of noise, and even with highly performing algorithms, the result of the extraction has to be manually checked for quality. With large volumes of data, this can be extremely costly. This is a known issue when working with comparable corpora (Zhang and Zweigenbaum, 2017). Yet, the issue is either not mentioned in works on parallel corpora creation from comparable corpora, or external information is used, such as metadata (Smith et al., 2010), which helps a lot the task.

In our work, we propose and evaluate methods for filtering out sentences and sentence pairs that have no chance of being of interest for the building of a parallel corpus. Hence, the purpose is to reduce the amount of manual check that needs to be performed on the output of a classifier.

2. Data collection and pre-processing

To perform our experiments, we work with a French comparable corpus containing biomedical documents with technical and simplified contents (Grabar and Cardon, 2018). The corpus is composed of three subcorpora: drug information for medical practitioners and patients released by the French Ministry of Health[1], medical literature reviews and their manual simplification released by the Cochrane foundation[2], and encyclopedia articles from Wikipedia[3] and Vikidia[4]. The documents are organised in pairs where the texts address the same topic for different audiences, so that the delivered information and the phrasing are not identical. More importantly, the order in which the information is delivered is not the same, which means that the document structure cannot be used for assuming where to look for parallel sentences.

For our experiments, we took 39 randomly selected document pairs from that corpus and manually annotated them for two types of sentence pairs :

- Equivalence : the sentences mean the same, but they are not identical;
- Inclusion : the meaning of one sentence is included in the other one, where additional information can also be found. This retains information about sentence splitting or merging and about information deletion or addition.

The documents are pre-processed for syntactic POS-tagging and syntactic analysis into constituents (Kitaev and Klein, 2018). In the manually annotated set, only sentences that have a verb are kept. This yields 266 sentence pairs: 136 equivalent pairs, and 130 inclusion pairs (56 in one direction, 74 in the other one).

For the automatic processing, we produced the whole possible combinations of sentences within each of the 39 document pairs, and ended up with 1,164,407 sentence pairs. Thus, given that, out of more than one million possible pairs, only 266 sentence pairs are considered as useful for the parallel corpus creation, we observe a high degree of imbalance: little less than 4,400:1. Our purpose is to reduce this imbalance for facilitating the search of parallel sentences and improving the overall quality of the results.

[1] http://base-donnees-publique.medicaments.gouv.fr/

[2] https://france.cochrane.org/revues-cochrane

[3] https://fr.wikipedia.org/

[4] https://fr.vikidia.org/

3. Method

In order to address that extremely high degree of imbalance, we propose to investigate three methods using formal and syntactic indicators:

- First method is based on the number of tokens in sentences. Hence, each candidate sentence must contain at least five tokens. This permits to consider sentences that are grammatically complete and convey some semantics. We set that value to five because that is the length of the shortest sentence in the set with the manual annotations;
- Second method prevents from producing pairs with identical sentences;
- Third method relies on syntactic information. We base our work on a method that uses constituency parsing for measuring similarity between sentences in a monolingual setting (Duran et al., 2014). In the original work, the authors detect similar words in sentences and assign a similarity score that is computed by looking at similar labels of nodes that contain similar words. The process is described in Figure 1. It is difficult to adapt that method as it is described in the paper. The main reason is that it relies heavily on a table that establishes which grammatical categories for constituents are similar to one another. It is made for English and there is no indication as to how it was built. Nonetheless, we make the assumption that adopting a similar approach could help in the process of weeding out undesired pairs for building a parallel corpus. Hence, instead of calculating a similarity score, we just choose between keeping the sentence pair as a candidate for a classifier, or rejecting it. For a given pair, we produce a syntactic tree for each of the two sentences. Then, if both sentences contain a verb, we compare all the leaves (i.e. words) of the trees, except the ones that are part of the stop words list. The list contains 83 items that are grammatical words, such as determiners or prepositions for example. If we find two identical words, we look at their parents nodes' labels. If those are identical, we keep the sentence in the candidates list. That process is illustrated in Algorithm 1 below. We also perform the same approach but instead of stopping if the parents nodes' labels are not identical, we go up a level to perform the same comparison, and up another level if the previous comparison was not successful. As soon as one comparison succeeds, we keep the sentence pair in the candidates list. This other approach is illustrated in Algorithm 2. That movement to the third parent of the leaves is what is chosen in the method which inspires this work, we chose to implement it to learn how the depth of exploration influences our filtering.

To parse the sentences in order to obtain their syntactic tree with constituents, we use the Berkeley Neural Parser and the language model that is provided with it for French, with the `benepar` Python library (Kitaev and Klein, 2018). The, we use the NLTK's `Tree` library (Bird et al., 2009) for tree manipulation and exploration.

Data: A pair of syntactic trees (T_1 and T_2), a list of stop words (*SW*)
Result: Boolean
Boolean ← False;
if *one verb is found in both sentences* **then**
  foreach *leaf in* T_1 *(*L_1*) not found in SW* **do**
    foreach *leaf in* T_2 *(*L_2*) not found in SW* **do**
      if L_1 *is identical to* L_2 **then**
        if L_1*'s parent node's label is identical to* L_2*'s parent node's label* **then**
          Boolean ← True;
        else
          nothing;
        end
      else
        nothing;
      end
    end
  end
else
  nothing;
end
return Boolean;

Algorithm 1: Filtering method only looking at the immediate parent nodes of the leaves

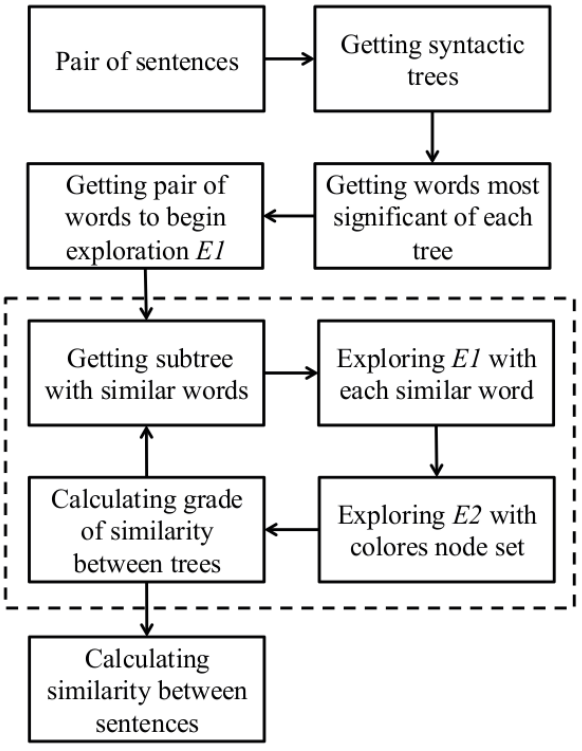

Figure 1: The similarity method described in (Duran et al., 2014)

4. Evaluation

We evaluate the results obtained in three different ways:

- we compare the number of initial sentence pairs to the number of remaining sentence pairs after the filtering,
- we check whether the removed pairs are manually an-

Remaining Pairs	*Unfiltered*	*FI*	*Syntax Depth 1*	*Syntax Depth 3*
Total	1,164,407	409,530	16,879	21,428
Equivalent	136	136	94	94
Inclusion	130	130	94	100

Table 1: Pairs remaining after the various filtering methods.

Data: A pair of syntactic trees (T_1 and T_2), a list of stop words (*SW*)
Result: Boolean
Boolean ← False;
if *one verb is found in both sentences* **then**
  foreach *leaf in T_1 (L_1) not found in SW* **do**
    foreach *leaf in T_2 (L_2) not found in SW* **do**
      if *L_1 is identical to L_2* **then**
        if *L_1's parent node's label (P_1) is identical to L_2's parent node's label (P_2)* **then**
          Boolean ← True;
        else
          if *P_1's parent node's label (PP_1) is identical to P_2's parent node's label (PP_2)* **then**
            Boolean ← True;
          else
            if *PP_1's parent node's label is identical to PP_2's parent node's label* **then**
              Boolean ← True;
            else
              nothing;
            end
          end
        end
      else
        nothing;
      end
    end
  end
else
  nothing;
end
return Boolean;

Algorithm 2: Filtering method looking up to the third parent node of the leaves

notated as parallel, be it equivalence or inclusion relation, in the reference dataset,

- we give the remaining data to a random forest classifier algorithm, such as done in a previous work (Cardon and Grabar, 2019), and evaluate recall and precision of the output.

The overall goal is to remove as many negative examples as possible, while preserving the positive examples.

5. Results and Discussion

We first look at how the volume of data is reduced further to the filtering operations. The first column in Table 1 shows the number of raw sentence pairs, the second colum indicates the number of pairs after using the formal indicators (FI), the third and fourth columns show the number of pairs remaining when using the syntactic filter, respectively with looking at the first syntactic parent node and up to the third parent node. The formal indicators are applied before the syntactic filters. The syntactic filters are used independently from one another.

We can see that the simple formal indicators reduce the total number of sentence pairs by 65% (from 1,164,407 to 409,530 sentence pairs). These two indicators were defined on the basis of observation of our data. They are very straightforward and we expected that no positive example (equivalent and inclusion pairs) would be lost in the process. This hypothesis is verified indeed: all the good candidates for parallel pairs are kept at this step.

Starting from the 409,530 pairs obtained after this first filter, we can see that both syntactic filters lead to a huge reduction of the volume of remaining sentence pairs:

- when using depth 1 leaves 16,879 pairs (∼96% reduction) remain,
- when using depth 3 leaves 21,428 pairs (∼95% reduction) remain.

The downside is that a substantial amount of positive examples is also lost in the process:

- 42 out of 136 (∼30%) for equivalent pairs with both depths used,
- 36 out of 130 (∼27%) for inclusion pairs with depth 1, 32 out of 130 (∼24%) for inclusion pairs with depth 3.

The over 95% reduction with the syntax filter on data that were already greatly reduced complies with our initial goal. Yet, we lose several good candidates for parallel sentences. Hence, we look at the positive examples that were rejected by the syntactic filter in order to understand why it is the case and how we can address this issue.

For instance, consider the following sentence pair:

- *Dans le cas où le patient devrait arrêter le traitement, il est recommandé de réduire progressivement la posologie. (In case the patient should stop the treatment, it is recommended to decrease the dose progressively.)*
- *L'arrêt du traitement doit se faire de manière progressive. (The cessation of treatment must be done progressively.)*

Set	*Unfiltered*			*FI*			*Syntax Depth 1*			*Syntax Depth 3*		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Equivalent Neg.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Equivalent Pos.	0.79	0.43	0.55	0.82	0.32	0.46	0.75	0.39	0.51	0.84	0.40	0.54
Inclusion Neg.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Inclusion Pos.	0.71	0.09	0.17	0.50	0.16	0.24	0.71	0.15	0.24	0.56	0.15	0.24

Table 2: Precision, Recall and F1 scores on the different sets of sentence pairs with classification.

The reason why this kind of sentence pairs is rejected is because the labels of parent nodes for identical words (such as *traitement (treatment)* in this example) differ in the trees produced by the syntactic parser. Indeed, in the first sentence, *le traitement (the treatment)* is labelled as an `NP-OBJ`, while it is labelled as an *NP* in the second sentence. The error is caused by the fact that *le traitement* from the second sentence (in *du traitement*, which is correctly analyzed as *de le traitement*) is an `NP` in a `PP` that depends on the noun *arrêt*. The parser that we use sometimes adds the information about the function of a phrase, this is the case in the first sentence here where *le traitement* is the object of the verb *arrêter*. This kind of examples suggests to put together similar node labels, such as `NP` and `NP-OBJ`. It would also be interesting to see whether some nodes are consistently similar in the parallel pairs, and hopefully find that those consistencies do not appear in pairs that should not be retained in a parallel corpus.
Let's analyze another typical example:

- *La prudence est recommandée chez les sujets atteints d'ulcères gastroduodénaux. (The vigilance is recommended in subjects suffering from gastroduodenal ulcers.)*
- *Ce médicament doit être utilisé avec prudence en cas d'ulcère de l'estomac ou du duodénum. (This medication must be used with vigilance in case of ulcers of the stomach and duodenum)*

There is only one pair of identical words here : *prudence (vigilance)*. This work is labelled as an `NP` in the first sentence and as a `PP` in the second sentence. The presence of *ulcère (ulcer)* in both sentences is not detected: the filter is currently looking for strictly identical words, while in these two sentences, *ulcère (ulcer)* occurs in its plural form in the first sentence and in its singular form in the second sentence. Hence, the filter must be more permissive in order to detect such occurrences. One solution is to work with a lemmatizer, another solution is to propose a more sophisticated word comparison function. This is a task where word embeddings could also be useful. We intend to test this possibility in future works.
Table 2 shows the results of classification with the different sentence pairs sets. For each experiment, the data were divided in two thirds for training and one third for testing. The results are reported by class (negative and positive) and positive class type (either equivalence or inclusion). The negative class has a perfect score in every metric because of the high degree of imbalance, the false negatives are not numerous enough to have an influence on the score. We can see that the syntactic method with a depth of exploration of three levels has a positive influence on precision, compared to unfiltered data, and recall is negatively impacted. We believe that being deprived of one third of such a small set of positive examples has a strong negative impact on performance. We should be able to improve recall if we prevent the positive examples from being filtered out, as we mentioned in the error analysis above. The results for inclusion show that this type of sentence pair is hard to recognize automatically. There is some improvement with filtered data, but the scores are low, especially recall. What we draw from those results is that the different sentence pairs types should be handled differently. It seems that we cannot expect to extract inclusion pairs in the same way as we extract equivalent pairs.

6. Conclusion

In this work, we proposed to address the problem of imbalance in the process of extracting parallel sentences from comparable corpora. We worked on a French comparable corpus made for biomedical text simplification. We showed that we could drastically reduce the number of negative examples (>98%) with simple heuristics and a syntactic comparison of sentence pairs, at the cost of losing some positive examples. Analyzing the errors, we showed that there were consistencies in what was left out and that should be kept, that can be addressed with improvements to the method, such as a better word comparison function and a more careful work on syntactic node label similarity. Even with those issues, we reduce the imbalance and improve precision on a classification task for equivalent sentences, thus reducing the manual work needed to check the output, which was the main objective. We also showed that inclusion pairs are much harder to process and that another method should be used for extracting that type.

7. Acknowledgements

We would like to thank the reviewers for their comments. This work was funded by the French National Agency for Research (ANR) as part of the *CLEAR* project (*Communication*, *Literacy*, *Education*, *Accessibility*, *Readability*), ANR-17-CE19-0016-01.

8. Bibliographical References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

Cardon, R. and Grabar, N. (2019). Parallel sentence retrieval from comparable corpora for biomedical text sim-

plification. In *Proceedings of Recent Advances in Natural Language Processing*, pages 168–177, Varna, Bulgaria, september.

Deléger, L. and Zweigenbaum, P. (2009). Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora (BUCC)*, pages 2–10, Singapore, August. Association for Computational Linguistics.

Duran, K., Rodriguez, J., and Bravo, M. (2014). Similarity of sentences through comparison of syntactic trees with pairs of similar words. In *11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–6, Campeche, 09.

Grabar, N. and Cardon, R. (2018). CLEAR – Simple Corpus for Medical French. In *Workshop on Automatic Text Adaption (ATA)*, pages 1–11, Tilburg, Netherlands.

Jhamtani, H., Gangal, V., Hovy, E., and Nyberg, E. (2017). Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark, September. Association for Computational Linguistics.

Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, July. Association for Computational Linguistics.

Smith, J. R., Quirk, C., and Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 403–411, USA. Association for Computational Linguistics.

Xu, W., Callison-Burch, C., and Napoles, C. (2015). Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Zhang, Z. and Zweigenbaum, P. (2017). zNLP: Identifying parallel sentences in Chinese-English comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 51–55, Vancouver, Canada, August. Association for Computational Linguistics.

Zweigenbaum, P., Sharoff, S., and Rapp, R. (2017). Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67, Vancouver, Canada, August. Association for Computational Linguistics.