Parallel Text Alignment and Monolingual Parallel Corpus Creation from Philosophical Texts for Text Simplification

Stefan Paun
University of Twente / Enschede, Netherlands
s.paun@student.utwente.nl

Abstract

Text simplification is a growing field with many potential useful applications. Training text simplification algorithms generally requires a lot of annotated data, however there are not many corpora suitable for this task. We propose a new unsupervised method for aligning text based on Doc2Vec embeddings and a new alignment algorithm, capable of aligning texts at different levels. Initial evaluation shows promising results for the new approach. We used the newly developed approach to create a new monolingual parallel corpus composed of the works of English early modern philosophers and their corresponding simplified versions.

1 Introduction

There has been a clear growth in research in the field of text simplification in recent years (Shardlow, 2014). Text simplification has many potential advantages, such as helping people who suffer from impairments like dyslexia (Alva-Manchego et al., 2020). Most recent approaches are data-driven and require learning text simplification transformations such as sentence splitting or word substitution from a parallel corpus.

Such a parallel corpus consists of a source document and a target document, which is the simplified version of it. The most widespread parallel corpora for text simplification are the parallel English Simple Wikipedia corpus (Zhu et al., 2010) and the more recent Newsela corpus (Xu et al., 2015).

A parallel corpus is obtained by aligning the units of text between the original-simplified pairs. The alignment can be done at different levels, however most research in the field is focused on sentence simplification (Alva-Manchego et al., 2020), thus sentence level alignments is the gold standard. Automated methods which can asses text similarity are highly desirable in order to produce such parallel corpora.

There are few tools that can easily align text in an unsupervised way. MASSAlign[1] by Paetzold et al. (2017) is a Python library which can produce alignments at both paragraph and sentence level in an unsupervised manner using a TF-IDF model. However, according to Campr and Ježek (2015), a Doc2Vec model would yield results which would imitate human estimates closer than a TF-IDF model when computing text similarity.

The current work has a twofold contribution. Firstly, we extend the existing MASSAlign tool with a Doc2Vec language model to better capture text similarity and a new alignment algorithm to complement the language model. We manually label two pairs of original-simpliefied documents and use these pairs to evaluate the performance of the Doc2Vec-based method. We find some promising results, however more evaluation is needed in order to draw a strong conclusion.

Secondly, we create a novel monolingual parallel corpus from philosophical texts. The novelty lies in the type of texts that constitute the corpus, specifically original philosophical works written by early modern English philosophers and their simplified variants re-written by a group of editors, with the scope to make the texts more accessible while preserving the meaning. The newly developed parallel corpus is created using the improved text alignment tool and is intended to be used as training data for existing text simplification systems. We sample alignments at random to get an idea of the quality of the corpus. Our initial findings show that the generated corpus seems to be of high quality.

2 Related Work

Paetzold et al. (2017) have proposed and developed an easy-to-use text alignment tool in the form of a Python library. Their approach relies on a simple TF-IDF model coupled with a Vicinity-Driven

[1] Github.com/ghpaetzold/massalign

Proceedings of NAACL-HLT 2021: Student Research Workshop, pages 40–46
June 6–11, 2021.

alignment method described in Paetzold and Specia (2016). Their alignment relies on the assumption that the order in which the information appears is consistent in both text pairs. Their system can identify one-to-many, many-to-one and many-to-many alignments, as opposed to the method proposed by Xu et al. (2015). This allows for capturing of text simplification operations such as splitting and compressing. Additionally, they employ a two stage approach, in which they first align paragraphs and then align sentences in the already aligned paragraphs. Our research expands and builds upon the work of Paetzold and Specia (2016).

Štajner et al. (2018) have presented CATS[2], a tool for the alignment of text simplification corpora. They employ two alignment methods, one which works under the same assumption as Paetzold et al. (2017), namely that the order of information is consistent in both pairs of text, and one which relaxes that assumption. Both approaches use the same strategy of aligning each sentence from the simplified-version of the document with the most similar sentence from original document, on the basis of textual similarity metrics. Similarly, their tool also allows for one-to-many and many-to-one alignments, and offers the option for a two staged alignment approach. One of their findings is that employing the assumption of consistent information ordering leads to an increase in the number of partial matches, at the cost of the number of full matches. However, this allows for the better capturing of the deletion operation specific to text simplification.

Xu et al. (2015) argue that the Simple Wikipedia corpus is a bottle neck for the text simplification field because the corpus is prone to automatic alignment errors, has inadequate simplifications and does not transfer well to other styles of texts. They present a new parallel corpus, Newsela, as an alternative to the Simple Wikipedia dataset. This new corpus improves on the shortcomings of the Wikipedia corpus since it consists of news articles professionally rewritten by editors. Our work provides an additional, novel corpus in order to advance the field of text simplification.

3 Dataset

The parallel dataset created is built from the works of four early modern philosophers, whose works were originally written in English: George Berkeley, David Hume, John Locke and John Stuart Mill. We obtained the original documents, which were in the public domain, from Project Gutenberg[3]. We obtained their simplified counter-parts of from Early Modern Texts[4]. The simplified version of texts were re-written by a team of editors, with the specific goal of making the original document more accessible while keeping the original ideas intact.

In order to be able to generate the parallel corpus, we cleaned-up and pre-processed the gathered data such that each document consists of a sentence per line, while empty lines represent the paragraph boundaries.

The pre-processing pipeline consists of multiple steps. First, using regular expressions we remove unwanted characters from the texts such as hashtags or underscores, or in the case of the simplified versions, characters that mark omissions or that are used for formatting purposes, which were added by the editors. The next step was to remove the newline characters found in the middle of sentences. This was also done by means of regular expressions. At the end of this step, the documents were formatted such that each line of the document represents a paragraph. Once this was achieved, a paragraph was split into sentences by using the Punkt Tokenizer provided in the NLTK[5] Python library. A list of common encountered abbreviations was supplied to the tokenizer such that sentences are not split midway.

4 Method

We use the open-source Python library, MASSAlign, developed by Paetzold et al. (2017) as the base for our new alignment algorithm. We expand the tool with a Doc2Vec language model and a new alignment algorithm which can take advantage of the new language model. Subsection 4.1 describes the language model, while Subsection 4.2 describes the alignment algorithm.

4.1 Language Model

Campr and Ježek (2015) evaluated a number of language models for the task of computing document similarity and found that TF-IDF embeddings are outperformed by Doc2Vec embeddings. This is in line with the intuition that a paragraph vector would capture meaning better than a simple bag of words

[2] Github.com/neosyon/SimpTextAlign

[3] Gutenberg.org

[4] EarlyModernTexts.com

[5] NLTK.org

approach since it makes better use of the context around words. Therefore, we decided to extend the MASSAlign tool with a Doc2Vec model.

The Doc2Vec model is used to create a vector embedding for the text unit to be aligned. In order to measure how similar two text units are, we use the cosine distance metric of the two vectors. We train a new Doc2Vec model each time an original-simplified pair of documents is to be aligned. The intuition behind is that this approach will better capture the specific style of the document.

We chose the parameters of the Doc2Vec model based on the insights from Lau and Baldwin (2016). Their empirical evaluation has shown that from the two methods employed by Doc2Vec, *dmpv* and *dbow*, the latter one yields better results, despite being less complex. They also find that instead of initializing word embeddings with random vectors, as it is typical with Doc2Vec, a step of *skip-gram* being performed before *dbow* leads to improvement in performance.

Therefore, the model is initialized with a vector size of 300, a window size of 15 and a negative sample of 5. The two parameters that are different from the findings of Lau and Baldwin (2016) are the number of training epochs and the minimum word count. Since some of the texts to be aligned are relatively short, a larger number of epochs and a smaller minimum word count is used in order to achieve more consistent results.

4.2 Alignment Algorithm

We developed an alignment algorithm to complement the Doc2Vec language model. The alignment algorithm is heavily inspired by the already existing Vicinity-Driven algorithm of (Paetzold and Specia, 2016). The need for another alignment algorithm was motivated by way the TF-IDF language model was used to determine whether two paragraphs are aligned. In the initial Vicinity-Driven method the similarity score of two paragraphs is given by the pair of sentences within the paragraph that have the highest similarity score. With the Doc2Vec model, a similarity score can be computed directly for the entire paragraph.

The new alignment algorithm starts from the beginning of the documents and looks for the first (original, simplified) pair of text units that are similar enough to consider. Once this candidate alignment pair is found, the next step is to try to improve alignment score, by expanding the initial alignment and looking for potential one-to-many and many-to-one alignments. Expanding the initial alignment is done by concatenating the current text units being considered with the next text unit from the original document, and, respectively, from the simplified document and computing new similarity scores. This expansion process continues until the newly computed similarity score stops improving. At this point the expansion process is stopped and the similarity score of this expanded candidate alignment pair is evaluated against a threshold. If the score is above the threshold, the candidate pair is considered aligned, otherwise, the algorithm looks for the next pair of text units which could be considered similar enough to try to align. The process continues until the end of both documents is reached. The algorithm allows for skipping of text units, to allow for the situation in which a particular text unit is not aligned to any text unit in the other document of the pair.

Similar to the original Vicinity-Driven method, the developed algorithm is capable of identifying one-to-one, many-to-one, one-to-many and many-to-many alignments. While it relies on the same assumptions as the original alignment algorithm, the approach described in this paper is able to relax one assumption, namely that the first paragraphs of the pair of documents are definitely aligned. Moreover, while the Vicinity-Driven approach employs two slightly different methods for aligning paragraphs and aligning sentences, the new method uses the same logic for both paragraph and sentence levels. This, coupled with the Doc2Vec model, makes the aligner capable of aligning text at different levels.

Unlike the already existing method, the new algorithm makes use of three different threshold levels. This is done for a number of reasons. First of all, a *certain threshold* is used to identify one-to-one alignments with a very high degree of similarity. A second, *hard threshold* is used to determine whether an alignment is good enough. A third threshold, *soft threshold* is employed in order to identify potential one-to-many, many-to-one or many-to-many alignments.

The thresholds are determined automatically by considering the distribution of the best similarity scores for each of the paragraphs or sentences of the simplified document from the initial similarity matrix. The *soft threshold* is determined by the lowest value of the similarity score distribution. Next, the 95% confidence interval where the

median value of the similarity score falls is determined. The *hard threshold* is determined by taking the lower boundary of the confidence interval and subtracting the standard deviation of the distribution, while the *certain threshold* is determined by considering the upper boundary of the confidence interval and adding the standard deviation of the distribution.

5 Results

5.1 Doc2Vec algorithm

In order to evaluate the performance of the proposed alignment algorithm, we have manually aligned two pairs of documents and created a ground-truth document for each pair of texts. The document which were used for evaluation are George Berkeley's "Essay Towards a New Theory of Vision" (Berkeley1709) and John Locke's "A Letter Concerning Toleration" (Locke1689b). We compared the performance of the original TF-IDF based Vicinity-Driven algorithm against the Doc2Vec based proposed algorithm.

Due to the statistical nature of Doc2Vec, running the alignment algorithm multiple times with the same parameters leads to small jitters in the results. The variation from run to run is determined by the quality of the Doc2Vec model, in particular for the number of epochs the model is trained. If the model is under-trained, there will be large variations in results between runs, thus it is important to have a model adjusted to the particularities of the text.

In order to evaluate the two methods, we consider the task of aligning sentences as a binary classification task, where each pair of sentences or paragraphs considered are either classified as correctly aligned or incorrectly aligned. We report the performance in terms of precision, recall and F1 measure. For sentences we consider two cases, one where the alignment is fully correct and one where the alignment is partial. In addition, we provide descriptive statistics about the one-to-one (1-to-1), many-to-one (n-to-1) and one-to-many (1-to-n) alignments. A one-to-many alignment implies that one unit of text from the original document maps to more than one unit of text from the simplified document, hence the original unit of text was split into multiple units in the simplified version.

The results are shown in Table 1. As it can be observed, for the Berkeley pair of documents, the Doc2Vec-based method seems to be slightly superior to TF-IDF, however the Doc2Vec-based approach performs worse in the case of the Locke pair of documents. Since the evaluation was performed on a limited sample of documents, there is not enough data to be able to infer anything categorically about the Doc2Vec-based approach.

	Berkeley		**Locke**	
	TF-IDF	Doc2Vec	TF-IDF	Doc2Vec
	Paragraph			
Detected	155	153	75	71
Correct	147	146	70	59
1-to-1	122	121	52	49
n-to-1	1	0	5	1
1-to-n	24	25	13	9
Precision	0.948	0.954	0.933	0.830
Recall	0.936	0.929	0.945	0.797
F1	0.942	0.941	0.939	0.813
	Sentences			
Detected	540	557	414	350
Correct	459	482	307	227
1-to-1	384	397	279	203
n-to-1	21	22	15	15
1-to-n	54	63	13	9
Precision	0.850	0.865	0.741	0.648
Recall	0.796	0.836	0.685	0.506
F1	0.822	0.850	0.712	0.568
	Partial Sentences			
Precision	0.948	0.935	0.908	0.797
Recall	0.888	0.904	0.839	0.622
F1	0.917	0.919	0.872	0.699

Table 1: Evaluation of TF-IDF model and original alignment algorithm against Doc2Vec model and our alignment algorithm for two pairs of documents

Table 2 contains examples which illustrate both successful and unsuccessful sentence alignments. Examples 1, 3.1 and 3.2 are from Berkeley's work (Berkeley1709), while examples 2 and 4 are from Locke's work (Locke1689b). Example 1 showcases a one-to-many type of alignment, in which the original sentence corresponds to two sentences from the simplified version. Example 2 showcases a many-to-one type of alignment, where two sentences of the original version correspond to a single sentence from the simplified document. Unsuccessful alignments can be classified as either partial or erroneous. With partial alignments there is some overlap between the original and simplified sentences, however the alignment fails to capture the full semantic similarity. A partial alignment can introduce offset in the alignment process and can

Ex.	Original Document	Simplified Document
	Successful alignments	
1	*to which i answer, it is not faintness anyhow applied that suggests greater magnitude, there being no necessary but only an experimental connexion between those two things.*	*i answer that what suggests larger size is not faintness as such but faintness of a kind and in circumstances that have been observed to accompany the vision of large sizes. we're not dealing with a necessary connection here, but only an experimental connection between those two things.*
2	*nay, we must not content ourselves with the narrow measures of bare justice; charity, bounty, and liberality must be added to it. this the gospel enjoins, this reason directs, and this that natural fellowship we are born into requires of us.*	*indeed, we should go beyond mere justice, adding benevolence and charity; the gospel commands this, reason urges it, and it is favoured by the natural fellowship we are born into.*
	Unsuccessful alignments	
3.1	*but, say you, the picture of the man is inverted, and yet the appearance is erect: i ask, what mean you by the picture of the man, or, which is the same thing, the visible man's being inverted?*	*you object: the picture of the man is inverted, yet the appearance is erect.*
3.2	*you tell me it is inverted, because the heels are uppermost and the head undermost?*	*what do you mean by the picture of the man? or, the same question, what do you mean by the visible man's being inverted? you tell me that it's inverted because the heels are uppermost and the head undermost?*
4	*another more secret evil, but more dangerous to the commonwealth, is when men arrogate to themselves, and to those of their own sect, some peculiar prerogative covered over with a specious show of deceitful words, but in effect opposite to the civil right of the community.*	*for if these were proposed thus nakedly and plainly, they would soon attract the attention of the magistrate and arouse the commonwealth to be on its guard against the spreading of such a dangerous evil.*

Table 2: Examples of successful and unsuccessful alignments.

cause the following sentence pair to also be only partially aligned, as illustrated by examples 3.1 and 3.2, which are consecutive pieces of text in the documents. With erroneous alignments, illustrated by example 4, the sentences convey different messages.

5.2 Parallel Corpus

The gathered documents have been aligned using the Doc2Vec aligner method. In Table 3 it is shown what percentage of the paragraph and sentence of the simplified documents have been aligned. This value gives an indication of how much of the document could be aligned, however it does not reflect the recall performance of the aligner since the total number of alignments will always be less or equal to the number of initial paragraphs or sentences, due to many to one alignments.

It can be observed that the coverage percentage is very low for larger documents. The cause of this is two-fold. Firstly, the Doc2Vec model is most likely under-powered since the hyperparameter values have been tuned on the *Berkeley1709* pair which is shorter than for instance *Berkeley1732*. Secondly,

	Paragraphs			**Sentences**		
Doc ID	Total	Det.	Cov.	Total	Det.	Cov.
Berkeley1709	157	154	0.98	576	547	0.94
Berkeley1710	185	173	0.93	1046	800	0.76
Berkeley1713	223	211	0.94	331	290	0.87
Berkeley1732	291	42	0.14	4228	865	0.12
Hume1739	1378	248	0.17	6687	865	0.12
Hume1748	277	114	0.41	1158	488	0.42
Hume1751	364	129	0.35	1348	422	0.31
Hume1779	264	254	0.96	1237	1140	0.91
Locke1689a	309	119	0.38	948	325	0.34
Locke1689b	88	71	0.80	616	350	0.56
Mill1843	1556	168	0.10	68686	426	0.06
Mill1859	140	124	0.88	1263	1109	0.87
Mill1863	111	91	0.81	696	602	0.86
Mill1869	96	85	0.88	1186	755	0.63
Mill1873	208	181	0.87	1879	1625	0.86

Table 3: Total and detected (Det.) paragraph (P) and sentence (S) alignments using Doc2Vec alignment method. Coverage (Cov.) shows the percentage of the total number of paragraphs and sentences that have been aligned.

by inspecting the documents with a low coverage, it was observed that there were a large number of short paragraphs and short sentences, of few words. These short paragraphs or sentences affect the performance of the Doc2Vec model since there is a lot less context when compared to longer paragraphs.

The alignments have been manually inspected by randomly sampling alignments from the different documents. While the sampling and inspection have not been performed in a structured manner, this was sufficient to determine that the text pairs which achieved a low coverage score were not optimally aligned. Therefore it would be detrimental to include these document pairs in the final corpus. Conversely, the text pairs which achieved a high coverage score appeared to be well aligned.

Therefore, we concatenated together the document pairs with a coverage value of above 0.3 to form a new corpus. Two files are created, for aligned paragraphs and for aligned sentences. The sentence alignment file consists of 8453 aligned sentences comprised of 636652 words in total. Another random sampling inspection is performed on the resulting corpus made of aligned sentences. Out of 100 sentence alignments extracted, 98 alignments can be classified as good, while 2 alignments can be classified as partial. A partial alignment means that there is an overlap between the aligned sentences, however, one of the sentence contains additional information which is not present in the other sentence.

6 Discussion

The current work has a number of limitations. One of the biggest limitations is that the evaluation of the performance of the Doc2Vec model is done with limited data points. While, it shows some promising results, the limited evaluation is not enough to allow for a strong conclusion to be drawn. To overcome this, a more extensive intrinsic and extrinsic evaluation should be performed by testing with parallel corpora that have already been aligned, such as the Simple Wikipedia corpus or the Newsela corpus and compare the number and quality of alignments obtained against already established methods.

In addition to a better evaluation of the model, a method for determining the hyperparameters of the Doc2Vec model based on the characteristics of the texts to be aligned, such as number of sentences or number of words, would be highly beneficial and would improve the alignment process in terms of both quality and time investment. Moreover, more recent, neural-network based language models, such as Sentence-BERT or Universal Sentence Encoder, could be considered as an alternative to Doc2Vec.

Another limitation of the current work is the lack of evaluation of the produced parallel corpus. While the limited random sampling shows very promising results, this is not enough in order to draw a conclusion regarding the quality of the resulted dataset. A more structured approach to the random sampling method could give better insight into the quality of the dataset.

Another point of improvement is the preprocessing stage. Ensuring that all text formatting elements, such as chapter numbers or titles are removed, would result in a more robust Doc2Vec model being trained on those documents. Moreover, very short paragraphs or sentences are detrimental to the quality of the Doc2Vec embeddings and do not add a lot of value for the text simplification process, thus they should be filtered out.

7 Conclusion

An approach to unsupervised text alignment was presented in this paper which makes use of Doc2Vec text embeddings in order to asses similarity between two pieces of texts. Additionally, an alignment method derived from the Vicinity-Driven approach of Paetzold and Specia (2016) has been presented. Initial results have shown the current work has slightly better performance compared to the original approach when evaluated on a specific pair of texts, but it has worse results on a different pair of texts. However, due to the limited evaluation, the outcome cannot be readily generalized and more testing is required in order to draw a definitive conclusion. The MASSAlign Python library has been extended to include this new Doc2Vec model.

A new monolingual parallel corpus has been created from documents consisting of works of English early modern philosophers and their simplified, corresponding, versions, which were redacted by a group of editors with the goal of making the original documents easier to follow and understand, while preserving meaning.

The newly created parallel corpus, together with the extended version of MASSAlign are available at: github.com/stefanpaun/massalign.

References

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Michal Campr and Karel Ježek. 2015. Comparing semantic models for evaluating automatic document summarization. In *Text, Speech, and Dialogue*, pages 252–260, Cham. Springer International Publishing.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany. Association for Computational Linguistics.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. MASSAlign: Alignment and annotation of comparable documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Tapei, Taiwan. Association for Computational Linguistics.

Gustavo Henrique Paetzold and Lucia Specia. 2016. Vicinity-driven paragraph and sentence alignment for comparable corpora. *ArXiv*, abs/1612.04113.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. CATS: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.