Automatic word alignment tools to scale production of manually aligned parallel texts

Stephen Grimes, Katherine Peterson, Xuansong Li

Linguistic Data Consortium, University of Pennsylvania
3600 Market Street, Suite 810, Philadelphia, PA, 19104, USA
{sgrimes, petka, xuansong}@ldc.upenn.edu

Abstract

We have been creating large-scale manual word alignment corpora for Arabic-English and Chinese-English language pairs in genres such as newsire, broadcast news and conversation, and web blogs. We are now meeting the challenge of word aligning further varieties of web data for Chinese and Arabic dialects. Human word alignment annotation can be costly and arduous. Alignment guidelines may be imprecise or underspecified in cases where parallel sentences are hard to compare — due to non-literal translations or differences between language structures. In order to speed annotation, we examine the effect that seeding manual alignments with automatic aligner output has on annotation speed and accuracy. We use automatic alignment methods that produce alignment results which are high precision and low recall to minimize annotator corrections. Results suggest that annotation time can be reduced by up to 20%, but we also found that reviewing and correcting automatic alignments requires more time than anticipated. We discuss throughout the paper crucial decisions on data structures for word alignment that likely have a significant impact on our results.

Keywords: word alignment, machine translation, annotation bootstrapping

1. Introduction

Manual word alignment corpora are word aligned parallel texts (often called bitexts). They are intended to serve as gold standard data for training and evaluation of automatic word alignment tools. Such tools in turn are essential in machine translation systems as orginally developed by (Brown et al., 1990). Word alignments are used in both phrase-based and syntax-based machine translation. Word alignments can also be useful in construction of bilingual dictionaries.

Manual word alignment can be an expensive, time consuming process, especially given the data volumes produced at organizations such as the Linguistic Data Consortium. A portion of the motivation of the current research is to control costs for this type of corpus development. Over a three-year period, annotators at the Linguistic Data Consortium (LDC) aligned approximately 2.1 million source tokens each of Arabic-English and Chinese-English parallel texts for the GALE program. This total includes word alignments on parallel treebanks of Arabic-English and Chinese-English of around a half million source tokens each (Li et al., 2010). This represents hundreds of hours of annotator time per week during peak periods. Hence finding ways to aid manual annotation represents a large potential cost savings. Eliminating routine, redundant annotation can also free up annotator attention for more challenging annotation scenarios.

Attempting to use automatic tools to bootstrap manual annotation is by no means a new notion in many areas of computational linguistics. For treebank annotation, parsers typically make an initial pass at analysis that is then corrected by manual annotation, such as when annotating the Arabic Treebank (Maamouri et al., 2008) or English Treebank (Marcus et al., 1993). Work has also been done to bootstrap parallel treebanks for German-Swedish (Volk and Samuelsson, 2004). The result is a feedback loop whereby the manual annotation trains the parser, enhancing its accuracy, which in turn aids in further corpus development. Examples of bootstrapping linguistic annotation in other areas of computational linguistics include semantic role labeling (Stevens, 2007) and sentiment analysis (Abdennadher et al., 2007). Indeed, this feedback loop method of bootstrapping annotation has been generalized to using machine learning to reduce data annotation time for many areas outside natural language processing (Schreiner et al., 2006).

Manual word alignment is an elaborate process; annotation guidelines can be dozens or hundreds of pages long. Human translators and in fact machine translation systems do not typically use words as the basis of translation but rather phrases, sentences, or larger units, and hence word alignment is often less natural than phrasal alignment. Due to structural differences between languages, one-to-one alignments are not always possible, leaving some words untranslated or standing in many-to-many correspondences. Hence, as will be demonstrated herein, even with dozens of pages of guidelines to instruct annotators, annotation agreement often tops out around 95% and can dip down to 85% for some genres. (Graca et al., 2008) report an average of 91.6% annotator agreement across several language pairs for their manual word alignment annotation.

The central question we attempt to address is whether using automatic alignments (we will term these “prealignments”) can produce a demonstrably faster manual annotation result, and if so by how much? At the outset of this process, it was unclear how annotators would respond to

the task of manual correction of automatic aligner output. Are the annotators more likely to simply accept default alignments provided to them at the expense of correctly interpreting annotation guidelines? How long does it take to review (and if necessary correct) prealignments versus word aligning clean parallel texts from scratch?

The paper is organized as follows. In Section 2. we begin by introducing several automatic aligners and their relative strengths and weaknesses. We must necessarily comment on the data structures used by the aligners; strangely, each data format is not capable of the capturing the same set of relationships and hence there do not exist bijective relationships between individual alignment formats. The distinct data formats affect measuring alignment performance, and in Section 4. we have related comments on measuring antator agreement in word alignment. In Section 5. we show the results of two small-scale test runs of our bootstrapping annotation on Chinese-English and Arabic-English parallel texts. Section 6. discusses the ongoing production run of this system for large scale alignment annotation and additional complications. Section 7. discusses future plans and concludes the paper.

2. Automatic aligners

GIZA++ (Och and Ney, 2003) has for several years been the baseline tool against which to compare all advances in word alignment. It implements IBM and HMM models. GIZA++ allows aligning one token from the source language to multiple tokens in the target language, i.e. one-to-many alignments, but does not allow multiple tokens from the source language to align to the same target token. Due to this asymmetry, running GIZA++ with source and target languages swapped produces different alignments. Because we desire a high level of precision for prealignments, we run GIZA++ twice, alternating the order of source and target languages, then take the fine intersection of the resulting alignments. The intersection necessarily contains only one-to-one alignments due to the restrictions of the GIZA++ structures. A variant of GIZA++ is MGIZA++, a derivative of GIZA++ which allows users to save trained model states.

The Berkeley Aligner (Liang et al., 2006) implements recent advancements in word alignment and allows both unsupervised and supervised use. In this context, supervised means that the aligner is trained with gold standard alignments. Unsupervised indicates that the training is only based on parallel corpora without alignments. It is an extension of the Cross-EM word alignmer. We found the Berkeley aligner useful because it allows for supervised training, enabling us to take advantage of previous corpora we have aligned.

We also considered using other automatic aligners. K-vec++ was an early implementation of the K-vec algorithm (Fung and Church, 1994). Unlike some other unsupervised algorithms, K-vec did not require that the input be sentence aligned, only that it be tokenized. The source and target documents are each divided into k partitions and each token is associated with a k-dimensional vector of binary values, with a 1 indicating the partitions in which the token occurs and a 0 indicating partitions in which it does not. Vectors for tokens from the source and target languages are compared using statistical measures of similarity, and tokens with highly similar vectors are aligned. K-vec did not support phrasal alignments, only correspondences between individual tokens. Fung and Church limited the K-vec algorithm to investigating words which occurred with a frequency between 3 and 10 in order to avoid considering too many pairs or pairs with two few occurences for statistical significance, but such bounds clearly do not scale to large data sets.

For both the Chinese-English and Arabic-English pilot prealignment, we used GIZA++ to generate word alignments, implementing IBM and HMM models which bootstrap one another. GIZA++ is run end-to-end twice, alternating ordering of the source and target languages. We then take the "fine" intersection of the proposed alignments. Section 4. has details about our distinction between "fine" and "coarse" alignment intersections. For large scale implementation of the prealignment method we use the Berkeley Aligner in order to leverage our existing manual annotations to increase automatic aligner accuracy.

The ITG model that implemented in the supervised variant of the Berkeley Aligner is resource intensive. The model grows with respect to the number of tokens per sentence. In practice, we have had little success using Berkeley Aligner with sentences longer than 40 tokens due to insufficient memory even after allocating memory to the Java virtual machine.

As a solution we have explored splitting longer sentences, running supervised alignment, and then concatenating the sentences back together. In the interim we limited our corpus to shorter sentences. Accordingly, reported F-scores don't reflect Berkeley Aligner's performance on the corpus as a whole and hence are artificially inflated. We are more concerned with generating high precision prealignments than achieving high recall, but the sentence length issue further decreases recall.

3. Word alignment data formats

Each word alignment corpus is produced differently depending upon input languages and annotation guidelines, but how the alignments are represented in a data format can also vary. Disregarding the ramifications of alignment structures may lead to poor performance, inappropriate design, and misinterpretations of others' work.

3.1. Basic alignment data structure

GIZA++ produces one-to-one or one-to-many alignments but it does not posit many-to-one or many-to-many relationships. In other words, GIZA++ strangely does not treat the two languages symmetrically, and this asymmetry is inherent in its data representation, i.e. even if the aligner were to posit a many-to-many alignment, the data format

cannot accommodate this.

The Berkeley Aligner (Liang et al., 2006) lists all alignments as one-to-one. However, unlike GIZA++, many-to-many representations can be inferred by post-processing alignments sharing common tokens to create many-to-many alignments. For instance, for tokens a_1, a_2 in language A and b_1 in language B, if we have a_1-b_1 (token a_1 is aligned to b_1) and a_2-b_1, then we can say equivalently that these two alignments are rather a single alignment, a_1,a_2-b_2.

The LDC Word Aligner allows for one-to-one or many-to-many alignments but with one caveat: our annotation tool and data release format stipulates that if a_1-b_1, a_1-b_2, and a_2-b_2, then necessarily it must be that a_2-b_1. In other words, if we consider a bipartite graph where tokens are vertices we require all connected components of the graph (alignments) to be completely connected subgraphs where all words are aligned to each other.

The LDC constraint requiring completely connected subgraphs has the side effect of enhancing annotator agreement. This is because once annotators have decided which tokens will comprise an alignment, they cannot choose anything but to completely link all component tokens of the alignment. By restricting choices agreement necessarily increases. Hence the LDC data type and GIZA++ share the property that not all alignment scenarios may be represented though they differ in the particulars of which alignments fail to be represented. Hence it is the case that the Berkeley data structure is the most general of the three described here.

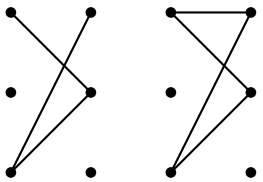

Figure 1: On the left: connected sub-components, not completely connected. On the right: completely connected.

3.2. Alignment link types

LDC word alignment corpora generally distinguish at a minimum between *correct* and *incorrect* links, where *correct* is selected by default and *incorrect* is used when the translation is inaccurate or not literal. Och and Ney (2003) distinguish between *sure* and *possible* links; *sure* links are links proposed by two or more annotators, while *possible* links only need to be proposed by at least one annotator. Yet because the names here are quite similar it is possible to confuse the two conventions, a dangerous mistake because an alignment marked as "incorrect" in our convention should not necessarily be considered "possible" in the convention Och and Ney use. Furthermore, in our convention, the link type labels "correct" and "incorrect" are determined by an annotator's judgment, and it would be problematic to mistake them for information about agreement between multiple annotators.

4. Measures of annotator agreement for word alignment

The word alignment community has not reached full consensus about how to measure accurarcy, and (Ahrenberg et al., 2000) provides a more-expansive discussion of some of the issues we address here. As a single statistic we prefer to use F-measure, which is now common to use instead of annotation error rate (AER) (Och and Ney, 2003). F-measure is defined as the harmonic mean of precision and recall:

$$\mathrm{F} = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

For bootstrapping manual word alignment with prealignments, it is important to increase precision at the expense of recall because recall, where annotators must correct an incorrect alignment, is more costly than creating an alignment where one did not previously exist. There is a certain human task of convincing annotators that prealignments can be reliable and accurate. Hence for this task we wish to optimize with respect to the $\mathrm{F}_{0.5}$ measure which weights precision more heavily than recall for the task. F_β is a generalization of F-measure and is be defined as:

$$\mathrm{F}_\beta = \frac{(1+\beta^2) * \mathrm{precision} * \mathrm{recall}}{(\beta^2 * \mathrm{precision}) + \mathrm{recall}}$$

In this equation beta represents how heavily recall is weighted relative to precision. The $\mathrm{F}_{0.5}$ measure thus weights recall as being half as important as precision in determining the score.

Because the LDC data format allows many-to-many alignments, we must explain how precision and recall are measured. Precision is the cardinality of the intersection of proposed alignments with gold-standard alignments divided by the cardinality of the gold-standard alignments. In the case of many-to-many alignments, if two alignments share constituents but are not identical, they do not appear in the intersection. However, if first many-to-many alignments are broken up into their component one-to-one alignments before the intersection is taken, the intersection is non-empty. We refer to these as coarse and fine intersections, respectively. Using fine intersections results in higher precision, recall, and F-measure. As systems only allowing one-to-one alignments by definition use fine intersection, we also use fine intersections for easy comparison.

In the context of annotation error rate, precision and recall definitions can be amended based on the notion of "sure" and "possible" links as discussed earlier. For precision, a proposed link is considered a match if it coincides with either a sure or possible link. In measuring recall, misses on possible links do not detract from the overall score because all annotators do not agree on possible links, by definition. We do not use these amended measure

for precision and recall because we do not use sure and possible, but it must be noted such conventions exist to avoid possible confusion.

5. Using automatic word alignment to seed manual alignment

To understand the effect of automatic alignments on manual annotation, it would be ideal to understand (a) how long an annotator must take to verify prealignments and (b) how long it takes to change incorrect prealignments. If either of the two above is too time consuming, any advantage gained by the annotator creating fewer alignments is offset.

5.1. Chinese experiment

Using approximately 200,000 tokens of Chinese newswire parallel text gathered as part of the GALE program as training for GIZA++, we produced alignments on four files of 1,000 source tokens each. (We specifically selected files which, when truncated to exactly 1,000 words, had the end of the file coinciding with a sentence boundary.) The precision of the prealignments produced by GIZA++ compared to gold standard annotation was high, ranging between 90-93%. As stated earlier, higher precision cannot be realistically expected, as this precision approaches an upper limit based on bounds on annotator agreement for word alignment that have been observed at the Linguistic Data Consortium and elsewhere (Graca et al., 2008). However, the recall score using this technique for GIZA++ is quite low, around 30%. The primary reason that recall is so low is that we take the intersection of forward and backward GIZA++ alignments (i.e. Chinese-English and English-Chinese, as directionality matters). As discussed earlier, this eliminates many-to-one and one-to-many alignments proposed by GIZA++ due to the nature of the GIZA++ data format. However, we have no desire to increase recall at the expense of precision because the penalty for correcting incorrect prealignments is high.

After creating prealignments on four files, we gave each of two annotators four files: two empty and two prealigned with GIZA++ alignments as described above. For the first annotator, files 1 and 4 were prealigned; for the second annotator it was files 2 and 4. Table 1 below gives the resulting F-measures between annotators and the previous two-pass annotations that we consider the gold standard.

As can be seen in the table above, annotator agreement on Chinese newswire texts attained results around 90%. Comparing human annotator agreement for our task on identical files (F-measure), it is observed that when both using GIZA++ prealignments agreement is at .91. The annotator agreement when not using GIZA++ prealignments is somewhat lower. However, it is premature to draw any conclusions regarding the effect of prealignments on annotator agreement at this time.

Turning to annotation speed, in Table 2 we see that in cases where one annotator started with a GIZA++ file and the other had a blank file, the person with the GIZA file was always faster, on average here by about 20%. We are cautious about this result and future trials with other annotators will determine how robust this measure is.

	Annotator 1	Annotator 2
File 1	.91 (G)	.91
File 2	.87	.91 (G)
File 3	.88	.88
File 4	.87 (G)	.86 (G)

Table 1: Annotator agreement using F-measure. (G) indicates files prealigned using GIZA++; otherwise files are manually aligned. In all cases, F-meausre is based on comparison to two-pass human files that were aligned as part of the GALE program and are treated as the gold standard here.

	Annotator 1	Annotator 2	Sentences
File 1	50 (G)	55	30
File 2	60	46 (G)	24
File 3	49	45	18
File 4	50 (G)	55 (G)	22

Table 2: Annotation times in minutes for four 1000-token Chinese files.

A 20% increase in speed is indeed significant, but we continue to strive for better results. We recognize that searching for and eliminating incorrect proposed alignments is also time consuming; overhead time is required to understand each sentence and assess prealigned tokens.

Note that File 3 was annotated quickly by both annotators. While each file contained exactly 1000 tokens, the number of sentences per file ranged from 18 to 30. This demonstrates a sentence effect — annotation speed is more closely correlated with the number of sentence segments than the number of tokens. We posit this is due to overhead required to understand each sentence. Once an alignment strategy is determined, tokens are aligned relatively quickly.

To further improve on our speed gains, we will train our automatic aligners on more same-genre data produced by LDC; this is the supervised approach and we use the Berkeley aligner to achieve these prealignments. It would also be useful to adopt approaches specific to Chinese-English language structures to improve alignment performance.

5.2. Arabic experiment

We only have preliminary data to report for the Arabic-English experiment. We ran GIZA++ on 164,984 Arabic tokens (corresponding to 201,031 English tokens) of Arabic-English parallel treebank text. The genre was broadcast news and was translated manually from Arabic.

The F-measure of the GIZA++ alignments to our gold-standard (two-pass manual annotation) was .602. In this case the GIZA++ intersection has almost identical precision and recall instead of the high precision, low recall that was found with Chinese. At issue is our data source: we used parallel treebanks including treebank tokens for alignment. Treebank tokens include empty category markers (e.g. syntactic traces) which are difficult for GIZA++ to match as they have no correspondent in the parallel language. We will re-tune our automatic alignment to obtain high precision, accepting lower recall as a result due to our focus on precision.

6. Large scale study

The results of the pilot experiments for Arabic-English and Chinese-English alignment were promising but were only based on annotation of 4000 tokens for each language pair. In some cases annotators may have been annotating files they had seen previously; another issue in the timing was that the annotators were specifically being asked to record times and give feedback on the prealignment files and they were not blind to details of our methods. To attain unbiased results, we will measure annotation speed through the course of large-scale corpus production over several months.

As part of the Broad Operational Language Technology program (BOLT), we are expecting to annotate several hundred thousand tokens. We feel that this data volume will be sufficient to assess average annotator speed. The BOLT data pose a particular challenge, however, because the genres for word alignment are web based: forums, email, instant messages, tweets, etc. Hence initially we may be unable to use our training data created during the GALE program; under GALE the genres were newswire, broadcast news and conversation, and web blogs. Also, under BOLT the language pairs we will initially manually align are Egyptian Arabic-English and Chinese-English. Hence we may be unable to use supervised training data and the Berkeley Aligner to produce our prealignments. We will instead opt to use GIZA++ on unsupervised parallel texts to produce prealignments, thereby reducing the F-measure of prealignment prior to manual annotation. After a certain threshold of data has been manually aligned from scratch in the new genres and language pairs, this data may be used to bootstrap supervised prealignments.

As discussed earlier, it is crucial to have annotators believe that the prealignments are helpful. If using unsupervised prealignments instead of supervised prealignments results in sufficient degredation of the prealignment quality, annotators may feel the prealignments are counterproductive. At best this may result in complaints but at worst annotators may opt to clear all prealignments in order to start annotation on a fresh file; this would render the prealignment program useless. Hence it is crucial to work with annotators, listen to their feedback, and demonstrate to them that the prealignments increase their annotation speed (when it is the case) in order to gain their trust and support.

7. Further research

There remains much work to be done to push the limits of how much annotation speed can be increased using prealignments without sacrificing quality. Of chief importance is ensuring that our automatic alignment techniques are as high precision as possible. We will continue to tweak combinations of alignment models in order to hone in on higher precision. At the Linguistic Data Consortium we are knowledgable in annotation — writing annotation GUIs, processing data, writing annotation guideless, etc. — but we are not experts in machine learning and models for machine translation. Increasingly we expect to consult with researchers at other institutions to learn about the latest developments in supervised and unsupervised word alignment model training.

In summary, we are encouraged by the modest results demonstrated thus far. We will continue to refine our automatic alignment techniques in an effort to produce quality gold standard alignments. If we are able to speed annotation and increase annotator agreement through these methods, this will increasingly allow annotators to devote their attention to more interesting annotation. The time savings will free up resources to create ever more diverse language resources.

8. Acknowledgments

This material is based upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-11-C-0145. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

9. References

Slim Abdennadher, Mohamed Aly, Dirk Bhler, Wolfgang Minker, and Johannes Pittermann. 2007. Becam tool-a semi-automatic tool for bootstrapping emotion corpus annotation and management. In *European Conference on Speech and Language Processing (EUROSPEECH)*, pages 946–949.

Lars Ahrenberg, Magnus Merkel, Anna Sagvall Hein, and Jorg Tiedemann. 2000. Evaluation of word alignment systems. In *Proceedings of LREC 2000*.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Pascale Fung and Kenneth Church. 1994. Kvec: A new approach for aligning parallel texts. In *Proceedings of COLING 94*, pages 1096–1102.

Joao Graca, Joana Paulo Pardal, Lusa Coheur, and Diamantino Antnio Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *The 6th International Conference on Language Resources and Evaluation, LREC 2008*.

Xuansong Li, Stephanie Strassel, Stephen Grimes, Safa Ismael, Xiaoyi Ma, Niyu Ge, Ann Bies, Nianwen Xue,

and Mohamed Maamouri. 2010. Parallel aligned treebank corpora at ldc: Methodology, annotation, and integration. In L. Ahrenberg, J. Tiedemann, and M. Volk, editors, *Proceedings of the Workshop on Annotation and Exploitation of Parallel Corpora*, Tartu, Estonia.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, and Seth Kulick. 2008. Enhanced annotation and parsing of the arabic treebank. In *Proceedings of INFOS*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguisticss*, 29(1):19–51.

Christopher Schreiner, Kari Torkkola, Mike Gardner, and Keshu Zhang. 2006. Using machine learning techniques to reduce data annotation time. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 2438–2442.

Gerwert Stevens. 2007. Xara: An xml- and rule-based semantic role labeler. In *Proceedings of the Linguistic Annotation Workshop*.

Martin Volk and Yvonne Samuelsson. 2004. Bootstrapping parallel treebanks. In Silvia Hansen-Schirra, Stephan Oepen, and Hans Uszkoreit, editors, *COLING 2004 5th International Workshop on Linguistically Interpreted Corpora*, pages 63–70, Geneva, Switzerland, Aug 29.