Natural Language Engineering **11** (3): 279–293. © 2005 Cambridge University Press
doi:10.1017/S1351324905003864 Printed in the United Kingdom

Optimization of word alignment clues

JÖRG TIEDEMANN
Alfa-Informatica, University of Groningen, Groningen, The Netherlands
e-mail: tiedeman@let.rug.nl

(*Received May 1 2004; revised November 30 2004*)

Abstract

Statistical, linguistic, and heuristic clues can be used for the alignment of words and multi-word units in parallel texts. This article describes the *clue alignment approach* and the optimization of its parameters using a *genetic algorithm*. Word alignment clues can come from various sources such as statistical alignment models, co-occurrence tests, string similarity scores and static dictionaries. A genetic algorithm implementing an evolutionary procedure can be used to optimize the parameters necessary for combining available clues. Experiments on English/Swedish bitext show a significant improvement of about 6% in F-scores compared to the baseline produced by statistical word alignment.[1]

1 Introduction

Word alignment is one of the basic operations when processing parallel corpora, with the aim of re-using for NLP tasks linguistic information stored in previously translated texts. Word alignment is used for extracting translation relations in computational lexicography (Karlgren, Karlgren, Nordström, Pettersson and Wahrolén 1994; Melamed 1996; Smadja, McKeown and Hatzivassiloglou 1996; Ribeiro, Lopes and Mexia 1901), terminology extraction (Dagan and Church:94), transfer-based, statistical and example-based machine translation (Brown, Della Pietra, Della Pietra and Mercer 1993; Vogel, Och, Tillmann, Nieen, Sawaf and Ney 2000; Dagan, church and Gale 1993; Ahrenberg 1999; Brown 1997). Additionally, word-sense disambiguation can be supported by cross-lingual links between words (Gale, Church and Yarowsky 1992; Diab and Resnik 2002) and semantic relations can be identified using aligned bitexts (Dyvik 2002). Lexical correspondences in bitexts can also be utilized for the adaptation of language tools to new languages (Borin 2002; Yarowsky, Ngai and Wicentowski 2001).

Automatic word alignment makes use of distributional properties and linguistic similarities between translation equivalents. Many types of features can be explored

[1] Most of the work described in this paper was carried out at the Department of Linguistics and Philology at Uppsala University. I would like to acknowledge technical and scientific support by people at the department in Uppsala.

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

in order to find indications for an association between word alignment candidates. The clue alignment approach (Tiedemann 2003a) incorporates different types of link indicators that can be combined for the identification of translation equivalents in parallel texts. One of the main challenges in this approach is the optimization of alignment parameters, which is the focus of this article.

Section 2 describes the clue alignment approach. Thereafter, we outline in section 3 our methodology for automatic evaluation of alignment results. Then, in section 4, we look at the optimization of alignment parameters using a genetic algorithm. Finally, experimental results are presented in section 5, followed by conclusions.

2 Clue alignment

Word alignment involves linking words or phrases in a source language text to their translations in a target language text. We base our approach on *word alignment clues*, i.e. pieces of evidence which are collected from different sources and combined in order to find the best solution for the given problem. The following two sections describe the notion of word alignment clues and how they can be used in the alignment of parallel corpora.

2.1 Word alignment clues

We distinguish between *declarative* clues which are pre-defined and *estimated* clues which are derived from the data analyzed. Declarative clues typically cover associations between words and phrases found in existing bilingual dictionaries. They may also include manually defined relations between similar linguistic features such as part-of-speech (Tiedemann 2003a). Estimated clues can be derived from co-occurrence measures such as the Dice coefficient (Smadja *et al.* 1996) or from string similarity measures such as the longest common sub-sequence ratio (LCSR) (Melamed 1995). Statistical alignment using the well-known IBM models (Brown *et al.* 1993) can also be applied to collect alignment clues.

Clues which are derived from different sources have to be weighted according to their importance. In general a clue is defined as the weighted association score S_i between features f_s and f_t of the two lexical items s and t: $C_i(s,t) = w_i S_i(f_s, f_t)$. The value of w_i is used to normalize and weight the score. Features can be any kind of information known about the two items and the context they appear in, e.g. their part-of-speech, the phrase type they occur in, or simply the lexical items themselves.

Thus, a word alignment clue $C(s,t)$ is used as a numerical indication of an association between two lexical items. More formally, it is defined as a function from pairs of lexical items s, t to the interval $[0,1]$ with the desired property that the probability $P(A(s,t))$ of an association $A(s,t)$ between these lexical items increases with increasing values of the clue: $c < c' \Rightarrow P(A(s,t) = 1|C(s,t) = c) < P(A(s,t) = 1|C(s,t) = c')$, where $A(s,t) = 1$ if s,t are associated and $A(s,t) = 0$ if they are not. Lexical items can be any set of words, multi-word units or single words.

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

2.2 Combining clues

Weighted alignment clues are used as independent indicators for a possible association between lexical items. Several association types can be found together, e.g. an association based on co-occurrence can be found together with an association based on string similarity. They can be combined to strengthen the evidence of translation relations indicated by different clues. The combination of all available clues for a given pair of lexical items is called the "total clue" (C_{all}). We combine clue scores in the following manner: $C_{all}(s,t) = C_1(s,t) \cup C_2(s,t) = C_1(s,t) + C_2(s,t) - C_1(s,t)C_2(s,t)$ (we define the intersection of two clues as simply the product of both: $C_1(s,t) \cap C_2(s,t) \equiv C_1(s,t)C_2(s,t)$). In this way, the total clue preserves the property of general alignment clues to produce values within the interval $[0,1]$. Indications of individual clues are added up to make the final decision. This implies that a single clue with the perfect score of 1.0 cannot be overruled by others. A stronger indication than this cannot be given because in this way we have one piece of evidence that provides a certainty of 100%. Intuitively, this should be sufficient to make a decision about the relation between the two lexical items. However, scores close to the perfect 1.0 should be exceptional because of their strong influence on alignment decisions. This should be accounted for by the clue weights defined for each of the clue types.

Note that not all types of clues have to be present to indicate a strong relation between lexical items. For example, the relation between cognates is usually indicated by a close similarity in spelling. However, translation relations between non-cognates can be as strong as for cognates but a clue based on string similarity cannot indicate that. A clue score combination based on the intersection rather than the union as we define it would give non-cognates an unfair penalty. Moreover, scores for certain types of clues will be zero for many word pairs. A combination of scores based on multiplication would cancel out all candidate pairs where at least one factor is zero even if they are translation equivalents and certain clues strongly suggests that they are. Leaving out zero-clues is not a solution to this problem either because this would punish candidates with more matching clues than others. On the other hand, using addition as defined above makes it possible to fairly compare combined clue scores that are derived from different numbers of individual clues.

Another important property is that clues do not have to refer to the same type of lexical items. Clues referring to multi-word units (MWUs) may overlap with clues referring to single words or other MWUs. A distributional clue based on co-occurrence frequency, for example, may suggest a link between the multi-word unit "The United Nations" and the Swedish compound "FN-konferensen" (*the United Nations conference*). Another one may indicate a relation between "Nations conference" and "FN-konferensen". Both clues refer to overlapping segments and both of them suggest incomplete links. However, both of them provide evidence for associations between these words in the bitext. In order to make use of such clues we define that clues indicate associations between all the words included in the segments they refer to. Using this definition we can compute combined clue values

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

```
English: The United Nations Conference has started today
Swedish: Idag började FN-konferensen
```

co-occurrence (DICE)		
The United Nations	FN-konferensen	0.3
Nations conference	FN-konferensen	0.4
has	började	0.2
started	började	0.6
started today	idag	0.3
string similarity (LCSR)		
conference	FN-konferensen	0.57
today	idag	0.4

	Idag	började	FN-konferensen
The	0	0	**30**
United	0	0	**30**
Nations	0	0	**58**
Conference	0	0	**74.2**
has	0	**20**	0
started	30	**72**	0
today	**58**	0	0

Fig. 1. Word alignment with clues, an example.

for each pair of words in a bitext using the clue combination rule as described above.

The example in Figure 1 illustrates the combination of overlapping clues in an English sentence and its translation into Swedish. Using the clues to the left of Figure 1, we can calculate values for the total alignment clue for each pair of words from the source and target language sentences. For example, two overlapping clues refer to the word pair "Nations" and "FN-konferensen". The combined value according to the clues in Figure 1 is then $C_{all}(\text{Nations, FN-konferensen}) = 0.3+0.4-0.3 * 0.4 = 0.58$. The same calculation can be done for each pair from the example. The resulting values can be represented in two-dimensional space, which we will call the *clue matrix* (right-hand side of Figure 1). The example matrix illustrates how overlapping clues may strengthen the evidence for certain alignments. The clue values indicated in bold mark links which would be found with a simple greedy search algorithm. Starting with the top score in the clue matrix, we can iteratively look for the next highest score until all words on both sides have been linked. In this way, alignments of multi-word units appear when links "overlap" with others as for example the links of (*The – FN-konferensen*), (*United – FN-konferensen*), (*Nations – FN-konferensen*) and (*Conference – FN-konferensen*). Putting these overlapping links together creates an alignment between (*The United Nations Conference*) and (*FN-konferensen*), which in this case is the correct alignment. In practice, several constraints can be added in order to avoid erroneous alignments, e.g. an adjacency constraint that restricts overlapping links to contiguous regions of the text (Tiedemann 2003a).

2.3 Statistical alignment models

Word alignment clues may be derived from any source that "promises" to find associations between words and phrases. Alignment approaches used in statistical machine translation (SMT) constitute one such source, producing, among other things, bilingual lexical translation probabilities. SMT uses refined translation models to estimate lexical probabilities that are directly applicable as a resource in the clue alignment system. In our experiments we use GIZA++ (Och 2003) for training statistical translation models. This software package implements IBM models 1 to 5 (Brown *et al.* 1993) with several refinements (Och and Ney 2000).

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

tokenized bitext	
Then hand luggage is opened.	Sedan öppnas handbagaget.
No one is very patient.	Ingen visar särskilt mycket tålamod.
3 initial characters ("token prefix")	
The han lug is ope.	Sed öpp han.
No one is ver pat.	Ing vis sär myc tål.
part-of-speech tags	
RB NNP NNP VBZ VBN.	RG0S V@IPSS NCNSN@DS FE
DT NN VBZ RB JJ.	PI@US0@S V@IPAS RG0S AQPNSNIS NCNSN@IS FE

Fig. 2. Example training data for statistical alignment (English/Swedish).

Statistical alignment models are usually trained on tokenized plain text files. It is also common to convert all tokens to lower case or lemma forms before running the training process. A corpus often contains additional linguistic information besides the plain text. Any kind of feature attached to or derived from words in the corpus may be used to train statistical alignment models. For example, a translation model can be trained on bitexts of part-of-speech tags or affixes taken from words in the corpus. Table 2 illustrates different types of training data[2] used to estimate statistical alignment models. Several features can be combined as well. For example, token prefixes and part-of-speech tags can be concatenated in order to represent items in the corpus.

Reducing the corpus to a sequence of part-of-speech tags or token affixes may be used to generalise the relations between source and target language tokens according to the selected features. In this way, data in the corpus is less sparse and therefore more appropriate for statistical investigations especially when using limited amounts of training data. Translation probabilities learned from different types of training data can be used as different types of alignment clues. Alignment results using these types are presented in section 5.

2.4 Bootstrapping

Alignment clues can be learned from word aligned training data. The process of clue learning is based on the assumption that an association clue C_j between lexical items is linearly correlated with the likelihood of f_s and f_t being features of aligned source and target language items. In this way, word alignment relations are captured in terms of generalized clues derived from data. *Clue patterns* can be defined such that they match features assumed to carry the relational information. There is a large variety of possible feature sets and many possible combinations among them. Some examples of clue patterns are listed in Table 1.

[2] The example bitext segments are taken from a translation of Saul Bellow's "To Jerusalem and back: a personal account" to Swedish (Bellow 1976) (henceforth the *Bellow* corpus (Sågvall Hein 2002)).

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

Table 1. *Feature patterns for learning clues.*

Name	Features (source and target)
dl	the word itself
dlp	word + its POS tag
dpx	POS tag + word position relative to the position of the aligned word
dp3	POS tag trigram (previous word, current word, next word)
dp3x	POS tag trigram (current, previous, next POS) + relative word positions
dc3x	chunk label trigram (current, previous, next chunk) + relative word positions

One way to calculate an association score for source and target language features from previously aligned training data is to use the Dice coefficient:

$$C_j(s,t) = w_j \frac{2 * P(f_s, f_t)}{P(f_s) + P(f_t)} = w_j * 2 * \left(\frac{1}{P(f_s|f_t)} + \frac{1}{P(f_t|f_s)} \right)^{-1}$$

According to our definition w_j is used to weight the Dice score in order to express the correlation between the relation of lexical items and the association between features attached to them. The Dice coefficient is especially useful because it is a symmetric combination of the two conditional probabilities $P(f_s|f_t)$ and $P(f_t|f_s)$.

Unfortunately, word aligned training data is not available and therefore, conditional feature probabilities cannot be estimated directly. However, other clues such as association measure clues may be used to create basic word alignments as described earlier. The idea now is to use automatically aligned data as training data for the estimation of feature clues. In this way, alignment clues can be found in a *bootstrapping procedure*, starting with a basic alignment and learning new clues from (noisy) alignments that have been produced using previous clues. Such *self-learning* techniques are known to increase noise. However, experimental results show that the alignment gains from clues learned dynamically (Tiedemann 2003a). The main reason for the success of this method is that previously unused information and contextual dependencies can be integrated in the alignment process in this way. For example, relations between part-of-speech tags can be found, which can be used to generalise the translation relation between words that belong to certain word classes. Dependencies between word positions can also be learned and syntactic relations can be identified. Clues learned from previously aligned bitexts will be called *bootstrapped clues.*

3 Automatic evaluation

Word alignment quality is usually measured in terms of precision and recall. Evaluation of alignments is usually carried out with respect to some previously created *gold standard* alignments. Gold standards can be re-used for additional test runs which is important when examining different parameter settings. However, recall and precision measures derived from information retrieval have to be adjusted for the task of word alignment. The main difficulty with these measures in connection with word alignment arises with links between MWUs that cause *partially correct* alignments. It is not straightforward how such links should be judged when

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

computing precision and recall. We use the following Q scores for partially correct links[3] (Tiedemann 2003b):

$$Q_x^{precision} = \frac{|alg_{src}^x \cap corr_{src}^x| + |alg_{trg}^x \cap corr_{trg}^x|}{|alg_{src}^x| + |alg_{trg}^x|}$$

$$Q_x^{recall} = \frac{|alg_{src}^x \cap corr_{src}^x| + |alg_{trg}^x \cap corr_{trg}^x|}{|corr_{src}^x| + |corr_{trg}^x|}$$

The set of alg_{src}^x includes all source language words of **all** proposed links if at least one of them is partially correct with respect to the reference link x from the gold standard. Similarly, alg_{trg}^x refers to all the proposed target language words. $corr_{src}^x$ and $corr_{trg}^x$ refer to the sets of source and target language words in link x of the gold standard. Using the value Q, we can define the recall and precision metrics as follows:

$$R_{mwu} = \frac{\sum_{x=1}^{X} Q_x^{recall}}{|correct|}, P_{mwu} = \frac{\sum_{x=1}^{X} Q_x^{precision}}{|aligned|}, F_{mwu} = \frac{2 * P_{mwu} * R_{mwu}}{P_{mwu} + R_{mwu}}$$

The evaluation measures proposed above are optimized for gold standards that have been sampled from a bitext. Links in the gold standard do not have to cover bitext segments completely and they may contain links between multi-word units.

4 Parameter optimization

The Clue Aligner allows the combination of any types of clues for the identification of links between words in the bitext. There are two questions to be addressed in this approach:

1. What are the most effective clues among all those that are available, and what is the optimal combination of clues for word alignment?
2. What are the optimal weights to be used for each clue in combination with others in order to maximize alignment precision and recall?

There are no trivial answers to either of these questions. There is a large variety of possible combinations and weighting schemes even for a small set of available clues. Misleading clues should be avoided or at the very least be weighted low. Strong indicators should be preferred and should obtain large weights. The weighting scheme should also account for the dependency between clues when used in combination. Unfortunately, there is no general numerical function that can be optimized in our model. However, automatic evaluation using gold standards as described above can be used to perform an approximate evaluation of alignment attempts. We can use this type of evaluation as the objective function for an iterative optimization procedure such as the one described below.

[3] $Q_x \equiv 0$ for incorrect links for both, precision and recall.

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

4.1 An evolutionary approach

Using the clue alignment approach and our methodology for automatic evaluation we want to model the optimization of clue sets and their weights as an evolutionary process using a genetic algorithm. The basic idea is to modify alignment settings step by step starting with the most basic ones. Using the gold standard we can evaluate alignment attempts and rank them according to their quality. Only the best attempts are developed further by modifying their parameters. The setting of the optimization process is as follows:

- Each clue alignment process is an *individual* in a population of processes. Each one of them is unique, characterized by its parameters. In our case, parameters are clue types and weights to be used for the alignment.
- Each of these individuals "grows up" by aligning a certain bitext (our training corpus). Their *fitness* is determined using the automatic alignment evaluation module and a gold standard for the underlying bitext.
- Only a small number of "grown-up" processes survive and may produce "children". These are the ones with the highest fitness scores in terms of alignment accuracy (F-value). This process is usually called *selection* in the terminology of genetic algorithms.
- Each living grown-up produces children as long as it lives. The clue alignment settings are inherited by its children. However, parameters are mutated for each of them. These children are the new generation in the evolutionary process and will compete with their parents when they are grown-ups.
- Mutation means any kind of modification of the settings inherited from the parent, e.g. additional clues or modified weights. Each setting (individual) is unique. Offspring with settings identical to previously existing individuals are not allowed (individuals with high fitness scores will stay in the population anyway and do not need to be re-produced).
- There is only room for a certain number of individuals to grow up at the same time. New offspring may not be produced until there is room for them.
- From time to time, the population learns clues from the currently strongest individual. New clues are directly available to the following generation. This is a very specific feature of our procedure based on the self-learning approach described in section 2.4.

This is in fact a very simple and straightforward setup for testing multiple clue alignment settings in a systematic way. There are several motivations for using a genetic algorithm for the iterative optimization of the clue aligner: Genetic algorithms repeatedly generate mutations of the current best hypotheses rather than searching linearly through the search space as in optimization using hill climbing techniques. In this manner the algorithm is less likely to get stuck in a local maximum. Individuals in each new generation may be much different from the parent generation, but only the most appropriate ones survive due to the selection mechanism. Similar to *simulated annealing*, settings are modified randomly and tested using an evaluation function. However, in genetic algorithms, several hypotheses are kept in parallel within the population from which new settings are derived. This

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

can be seen as a "generate-and-test beam search" (Mitchell 1997) through the space of hypotheses. From this also follows that genetic algorithms are easily parallelized where individuals of the current population develop simultaneously. Implementation issues about parallel processes are discussed further down.

4.2 Optimization parameters

The approach described above requires several initial parameters. First, we have to decide what kind of clues are available for the individuals in the population. It could even be possible to vary some of the clues provided to the system. For instance, clue patterns as described in 2.4 could be mutated. However, this is not the subject of the present study. We will focus on a fixed set of clue types.

Secondly, we have to decide what mutation operations are possible when a new individual is born. Here, we include six basic operations: 1) adding one clue type that is not included yet; 2) increasing one of the clue weights by a fixed value; 3) decreasing one of the clue weights by a fixed value; 4) increasing the threshold for the total clue scores by a fixed value; 5) decreasing threshold for the total clue scores; 6) removing a randomly chosen clue type. When a child is born, mutation operations are repeatedly applied until its setting is unique. The "clue-adding-operation" is preferred to the "weight-changing" operations. Clues to be added or removed are chosen at random. New clues are weighted using a fixed initial clue weight. Mutation preferences and weight modification values are arbitrarily chosen and are fixed throughout the optimization process.

Note that we only apply the pre-defined mutation operations to create new individuals from exactly one parent. In other genetic algorithms, crossover operations are usually performed to combine the parameters of two randomly chosen parents to create a new child. Mutation is then additionally applied to only a small portion of individuals whereas we use mutation with all of them.

Another parameter controls the intervals between the self-learning periods as described in section 2.4. In our setup, self-learning is triggered each time a certain number of new individuals has "grown up" in the entire population, i.e. each time a certain number of alignment processes has finished. Bootstrapped clues are always learned from the currently best performing individual.

The last initial parameter is the size of the living space within the population. In our settings, we simply define a fixed number of individuals who may live simultaneously. First, only a certain number of "grown-ups" may survive. Only the strongest ones will survive; all the others will die. Further, only a certain number of children may grow up at the same time.

4.3 Implementation issues

The evolutionary optimization process has been implemented as a server application that co-ordinates the development of the population by running client processes on the local network. Initially, basic clues are produced by the server itself. The next step is to collect available clue types and to start the evolutionary loop. Alignment processes (growing individuals) are queued to be processed by any client on the

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

network. They are distributed using a simple client-server architecture. Processes in the queue are basically distributed to appropriate clients that request new tasks.

Each client process includes an evaluation process in order to produce "fitness" scores from the alignment result. The server keeps track of running clients and collects evaluation scores when they are available. Child processes are placed in the queue as long as there are free slots in the living space. Note that in our setting children that grow fast have a "natural advantage" in propagating compared to the slow growing ones. In this way, alignment processes that take less time have a larger chance to be further developed than slower ones. However, they and their offspring may still die out if a slower one reports a better result later on. In the selection phase, competing individuals are always the current "grown-ups". New children enter this competition as soon as they are finished and report their fitness scores.

The process queue is handled independently from the optimization process. Client processes in the queue are served as soon as there is a slot available. Preference mechanisms are not implemented. There are usually more processes in the queue than can be run simultaneously. Clients in our network are only available if nobody else is working with the client at the same time. For our experiments we used a local network of GNU/Linux clients used by staff and students at the department. The number of available clients in our settings varies between 0 and 29 depending on the time of the day and course schedules. This introduces an additional randomness to the optimization procedure.

5 Experiments

We evaluated our approach using the English/Swedish Bellow corpus (Sågvall Hein 2002). This corpus is fairly small (about 140,000 words with about 4200 sentence pairs) and therefore well suited for extensive studies of alignment parameters. The fitness scores during the evolutionary optimization are measured using a manually made gold standard including 456 links (Merkel, Andersson and Ahrenberg 1999). A second distinct gold standard of 875 manually aligned links is used for evaluation. The sets are disjoint but have been produced using the same procedures and guidelines (Merkel 1999). The Bellow corpus has been tagged and parsed using the Grok system for English (Baldridge 2002), the TnT tagger (Brants 2000; Megyesi 2002) and the CFG parser for Swedish (Megyesi 2002). The following clue types have been applied:

Basic clues: The Dice coefficient and LCSR scores for co-occurring words and chunks.

SMT clues: Translation probabilities derived from statistical translation models trained on various kinds of training data using GIZA++:

- tokenized plain text (*giza-word*)
- part-of-speech tags (*giza-pos*)
- plain text and part-of-speech tags (*giza-word-pos*)
- token prefixes (three initial characters) (*giza-word-pref*)
- token suffixes (three final characters) (*giza-word-suf*)

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

The training process was run in both directions (source tokens aligned to target tokens and vice versa). This gives a total of 10 different SMT clue types (inverse alignments will be marked with *'_i'*).

Bootstrapped clues: Clues learned from previously word-aligned data, with feature patterns as listed in Table 1 (*dl, dlp, dpx, dp3, dp3x, dc3x*).

Mutation in the optimization process uses the six operations described in section 4.1. The evolutionary process starts with individuals that use single clue types only (taken from the list of available clues; bootstrapped clues are not available initially). The number of individuals growing up at the same time has been limited to 20. The living space for "grown-ups" has been set to a maximum of 10 individuals. New offspring are produced by copying the existing settings and modifying them using the mutation operations. The following arbitrarily chosen parameters are used: The probability of applying the clue-adding-operation is set to 0.5. All possible clue types have the same chance of being added when this operation is applied. The probability of changing a randomly selected weight is set to 1/6 (equally for increasing and decreasing the weight), the probability of changing the threshold for the total clue score is set to 2/30 (equally for increasing and decreasing the threshold), and the probability of removing a clue type is set to 1/30. The default clue weight is set to 0.05. Weights and thresholds are changed using a fixed value of 0.01. Mutation operations are continuously applied until a new unique setting is found that has never been tried before.

The time for learning clues is determined by the total number of grown-up individuals that have been part of the population, i.e. by the number of alignment scores collected by the server application. In our experiments, bootstrapped clues are learned each time 100 new scores have been stored in the evaluation table.

5.1 Baseline

We used various baselines in order to compare the results achieved by the clue aligner using different settings. One result of training statistical alignment models with GIZA++ is the final Viterbi alignment, i.e. the optimal alignment between words in the bitext according to the final translation model. GIZA++ can be run in two directions, aligning target language words to source language words (*GIZA++*) and aligning source language words to target language words (*GIZA++ inverse*). Both Viterbi alignments were evaluated using the same gold standard as used in all other alignment experiments. They can also be combined using heuristics as proposed in, for example, Och and Ney (2000).

5.2 Results

Baseline results are presented in Table 2 along with the evolution of F-scores for a population of 700 individuals. The development of F-scores in the evolutionary process is non-monotonic as can be seen in Figure 3, due to the random variation in the mutation process. However, the overall picture shows gradual improvement over time. The top score of 89.9% in F-value for the training data is clearly an

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

Table 2. *Alignment results*

	Train	*Test (875 unseen links)*		
baseline	**F**	**precision**	**recall**	**F**
GIZA++	78.8%	74.1%	76.2%	75.1%
GIZA++ inverse	83.0%	79.0%	81.8%	80.4%
GIZA++ union	82.6%	70.3%	93.8%	80.4%
GIZA++ intersection	83.4%	91.5%	69.7%	79.1%
GIZA++ refined	83.1%	82.9%	77.7%	80.2%
evolution	**F**	**precision**	**recall**	**F**
lowest (LCSR only)	25.1%	35.3%	22.9%	27.8%
highest (after 100)	80.1%	76.8%	83.4%	80.0%
highest (after 200)	85.5%	79.3%	87.0%	82.9%
highest (after 300)	87.0%	83.0%	85.1%	84.0%
highest (after 500)	89.3%	84.9%	85.9%	85.4%
highest (after 700)	89.9%	85.7%	86.6%	86.1%

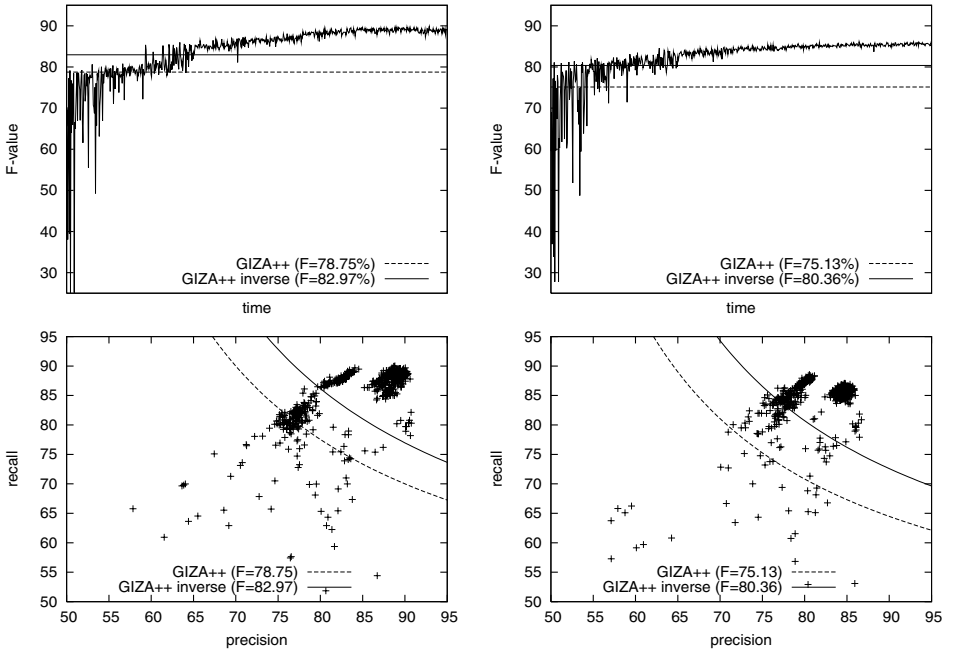


Fig. 3. Parameter optimization. Training (left) and testing (right).

improvement also compared to the baseline alignments. It was produced using GIZA++ clues (giza-word, giza-word-i, giza-pos, giza-word-pref and giza-word-pref-i), Dice scores, and the following bootstrapped clues: *dc3x*, *dlp* and *dp3*.

The correlation between scores retrieved in the training process and their corresponding scores in the evaluation can also be seen in Figure 3. The charts at the bottom of the figure illustrate the development of the population in terms of precision and recall values in training and evaluation. Here, it can be seen how

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

Table 3. *Clue type distribution for 700 settings, 511 of which include bootstrapped clues*

clue	*used*	*%*	*clue*	*used*	*%*	*clue*	*used*	*%*
giza-word	584	83.4	LCSR	166	23.7	dc3x	492	96.3
giza-word-i	578	82.6	giza-word-pref	161	23.0	dlp	344	67.3
giza-word-pref-i	524	74.9	giza-word-pos	122	17.4	dp3	259	50.7
giza-pos	429	61.3	giza-word-suf	88	12.6	dl	145	28.4
Dice	217	31.0	giza-pos-i	64	9.1	dp3x	58	11.4
giza-word-suf-i	213	30.4	giza-word-pos-i	55	7.9	dpx	46	9.0

the population and its individuals move to a space of "improved fitness". The optimization is certainly biased towards the training data but the evaluation scores suggest a strong correlation to results on unseen test data.

Table 3 shows the distribution of clue types used by individuals in the optimization process. GIZA++ clues have been used in most of the settings. Some of them seem to be less important than others, e.g. clues derived from POS tag alignments and alignments of suffix characters are less frequently selected. Surprisingly, the GIZA++ alignment using a combination of words and POS tags is not used very often. This is probably due to the size of the corpus and the sparseness of data when training on complex data. It would be interesting to see the effect of larger training data on the selection of GIZA++ clues. Furthermore, Dice and LCSR clues are not selected very often either. LCSR scores in particular perform poorly even on closely related languages such as English and Swedish. These clues seem to provide only very little or possibly even confusing evidence to the aligner. Among the bootstrapped clues, the chunk-label-position clue seems to perform best. It is chosen in almost all settings that include any bootstrapped clue. Lexical clues, especially in combination with POS tags, are also frequently selected. This is probably due to the high precision of these clues learned from previous alignments. Other bootstrapped clues such as POS-position clues seem to be too general and therefore the aligner seems to avoid them. On the other hand, clues based on POS-trigrams are selected very frequently. The randomness of the genetic algorithm makes it hard to draw general conclusions from these figures. It is difficult to formulate a stop criterion, although the process seems to converge after 600–700 individuals tested. However, the optimization process is still proceeding in small steps and a step backwards does not necessarily mean that optimization is taking a "wrong turn". Furthermore, the definition of mutation operations for producing new generations is crucial for our procedure. Additional experiments with varied settings may show more clearly the tendencies in clue selection.

6 Summary and conclusions

This article has presented the clue alignment approach to automatic word alignment and the optimization of its parameters using an evolutionary approach. The clue aligner allows one to combine evidence from different sources to improve alignment

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

quality. Alignment patterns can also be learned from automatically word-aligned data. The variety of parameters in such an approach can be optimized using genetic algorithms. The optimization process presented in this article makes use of automatic evaluation using a gold standard and a simple propagation of alignment settings in an evolutionary process. In this way, the aligner is trained according to the reference links in the gold standard. We tested this algorithm with an English/Swedish bitext and several alignment clues. A second gold standard with unseen links has been used for evaluation. The optimization resulted into an improvement of about 6% in F-value after 700 tested settings compared to the best baseline alignment produced by statistical word alignment.

References

Ahrenberg, L. (1999) Superlinks: A new approach to containing transfer in machine translation. Technical report, Linköping University, Linköping, Sweden.

Baldridge, J. (2002) Grok – an open source nlp library. http://grok.sourceforge.net/.

Bellow, S. (1976) *From Jerusalem and Back: A personal account*. The Viking Press, New York, USA. Translation to Swedish by Caj Lundgren, Bonniers, Stockholm, 1977.

Borin, L. (2002) Alignment and tagging. *Parallel Corpora, Parallel Worlds: Proceedings of the Symposium on Parallel Corpora*, pp. 207–218. Rodopi, Amsterdam, New York.

Brants, T. (2000) TnT – a statistical part-of-speech tagger. *Proceedings of the 6th Applied Natural Language Processing Conference ANLP-2000*, pp. 224–231. Seattle, WA.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J. and Mercer, R. L. (1993) The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* **19**(2): 263–311.

Brown, R. D. (1997) Automated dictionary extraction for 'knowledge-free' example-based translation. *Proceedings of the 7th International on Theoretical and Methodological Issues in Machine Translation, TMI-97*, pp. 111–118. Santa Fe, NM.

Dagan, I. and Church, K. W. (1994) Termight: Identifying and translating technical terminology. *Proceedings of the 4th Conference on Applied Natural Language Processing*, pp. 34–40. Stuttgart, Germany.

Dagan, I., Church, K. W. and Gale, W. A. (1993) Robust bilingual word alignment for machine-aided translation. *Proceedings of the Workshop on Very Large Corpora*, pp. 1–8, Columbus, OH.

Diab, M. and Resnik, P. (2002) An unsupervised method for word sense tagging using parallel corpora. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 255–262. Philadelphia, PA.

Dyvik, H. (2002) Translations as semantic mirrors: From parallel corpus to Wordnet. *23rd International Conference on English Language Research on Computerized Corpora of Modern and Medieval English (ICAME)*, Gothenburg, Sweden.

Gale, W. A., Church, K. W. and Yarowsky, D. (1992) Using bilingual materials to develop word sense disambiguation methods. *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 101–112. Montreal, Canada.

Karlgren, H., Karlgren, J., Nordström, M., Pettersson, P. and Wahrolén, B. (1994) Dilemma – an instant lexicographer. *Proceedings of the 15th International Conference on Computational Linguisticss (COLING)*, Kyoto, Japan.

Megyesi, B. (2002) Shallow parsing with POS taggers and linguistic features. *Journal of Machine Learning Research*, Special Issue on Shallow Parsing, **2**: 639–668.

Melamed, I. D. (1995) Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In: D. Yarovsky and K. Church, editors, *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 184–198. Boston, MA.

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press

Melamed, I. D. (1996) Automatic construction of clean broad-coverage lexicons. *Proceedings of the 2nd Conference the Association for Machine Translation in the Americas (AMTA)*, pp. 125–134. Montreal, Canada.

Merkel, M., Andersson, M. and Ahrenberg, L. (2002) The PLUG link annotator – interactive construction of data from parallel corpora. In: L. Borin, editor, *Parallel Corpora, Parallel Worlds: Proceedings of the Symposium on Parallel Corpora*. Rodopi, Amsterdam, New York.

Merkel, M. (1999) Annotation style guide for the PLUG link annotator. Technical report, Linköping University, Linköping, Sweden.

Mitchell, T. M. (1997) *Machine Learning*. McGraw Hill.

Och, F. J. and Ney, H. (2000) Improved statistical alignment models. *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 440–447.

Och, F. J. (2003) GIZA++: Training of statistical translation models. Available from `http://www.isi.edu/~och/GIZA++.html`. (new version).

Ribeiro, A., Lopes, G. and Mexia, J. (2001) Extracting translation equivalents from Portuguese-Chinese parallel texts. *Journal of Studies in Lexicography* **11**(1): 118–194.

Hein, A. S. (2002) The PLUG project: Parallel corpora in Linköping, Uppsala, and Göteborg: Aims and achievements. In: L. Borin, editor, *Parallel Corpora, Parallel Worlds: Proceedings of the Symposium on Parallel Corpora*. Rodopi, Amsterdam, New York.

Smadja, F. A., McKeown, K. R. and Hatzivassiloglou, V. (1996) Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics* **22**(1): 1–38.

Tiedemann, J. (2003a) Combining clues for word alignment. *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 339–346. Budapest, Hungary.

Tiedemann, J. (2003b) *Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD thesis, Department of Linguistics, Uppsala University, Uppsala/Sweden.

Vogel, S., Och, F. J., Tillmann, C., Nieen, S., Sawaf, H. and Ney, H. (2000) Statistical methods for machine translation. In: W. Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 377–393. Springer Verlag.

Yarowsky, D., Ngai, G. and Wicentowski, R. (2001) Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of the 1st International Conference on Human Language Technology Research (HLT)*.

https://doi.org/10.1017/S1351324905003864 Published online by Cambridge University Press