Jody Foo
jodfo@ida.liu.se

An overview of bitext alignment algorithms

1. Background

Parallel corpora (bitexts) are interesting to study in many fields. In translation studies they can be used to study how translators work. In bilingual lexicography, they can be used to discern new lexicograhic patterns, and in machine translation, especially statistical machine translation, they can be used to create statistical resources such as translation models. Word aligned corpora can also be used in term extraction. In machine translation, phrase and word level alignment is of most interest as they are used to create translation models.

2. Survey of alignment methods

Text alignment can be done at many levels, ranging from document alignment to character alignment with , paragraph, sentence, and word alignment in between.

In most literature, alignment methods are categorized as either statistic or heuristic approaches. Statistic approaches estimate alignment probabilities whereas heuristic approaches use associative measures derived either from corpora, or external sources such as dictionaries.

2.1. Statistical Alignment Models

2.1.1 Gale and Church, 1993

Based on the recent interest in studying bilingual corpora, Gale and Church published a paper in 1993 (Gale & Church, 1993) that described a program and a method of aligning sentence units in a bilingual corpora. The method is based on character based sentence length correlations, i.e. "*the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences*". Gale and Church assign a probabilistic score to proposed sentence pairs based on a distance measure δ. Of the proposed sentence pairs, the most likely proposal is selected using the *maximum likelihood* algorithm.

The sentence lengths are assumed to follow a normal distribution. The mean character length ratio of sentences in L1 compared to sentences in L2 is assigned to the variable c. The variance of this distribution is assigned to s. For each sentence pair, l1 and l2, the length difference δ is calculated as follows

$$\delta = (l_2 - l_1 c) \Big/ \sqrt{l_1 s^2}$$

Equation 1.

This length difference metric δ is used by Gale and Church to create a conditional probabilistic model which they then use to estimate sentence alignment likelihood. The alignment is found by choosing the generated alignment that has the highest likelihood.

2.1.2 The IBM Alignment Models 1 through 4

In their systematic review of statistical alignment models (Och & Ney, 2003), Och and Ney describe the essence of statistical alignment as trying to model the probabilistic relationship between the source language string f, the target language string e, and the alignment a between positions in f and e. The mathematical notations commonly used for statistical alignment models follow.

$$f_1^J = f_1, ..., f_j, ..., f_J$$
$$e_1^I = e_1, ..., e_i, ..., e_I$$

Equation 2.

Foreign and English sentences f and e, contain a number of tokens, J and I (Equation 2). Tokens in sentences f and e can be aligned, correspond to one another. The set of possible alignments is denoted $\mathcal{A}$, and each alignment from j to i (foreign to English) is denoted by a_j which holds the index of the corresponding token i in the English sentence (see Equation 3).

$$\mathcal{A} \subseteq \{(j,i): j = 1, ...J; i = 1, ..., I\}$$
$$j \rightarrow i = a$$
$$i = a_j$$

Equation 3.

The basic alignment model using the above described notation can be seen in Equation 4.

$$\Pr\left(f_1^J \mid e_1^I\right)$$
$$\Pr\left(f_1^J, a_1^J \mid e_1^I\right)$$
$$\Pr\left(f_1^J \mid e_1^I\right) = \sum_{a_1^J} \Pr\left(f_1^J, a_1^J \mid e_1^I\right)$$

Equation 4.

From the basic translation model $\Pr\left(f_1^J \mid e_1^I\right)$, the alignment is included into the equation to express the likelihood of a certain alignment mapping one token in sentence f to a to-

ken in sentence e, $\Pr\left(f_1^j, a_1^j \mid e_1^l\right)$. If all alignments are considered, the total likelihood should be equal to the basic translation model probability.

The above described model is the *IBM Model 1*. The model has been improved since its inception and the successors have been named Model 2, Model 3, Model 4, and Model 5.

Model 2

One problem of Model 1 is that it does not have any way of differentiating between alignments that align words on the oposite ends of the sentences, from alignments which are closer. Model 2 adds this distinction.

Model 3

Languages such as Swedish and German make use of compound words. Languages such as English do not. This difference makes translating between such languages impossible for certain words, the previous models 2 and 3 would not be capable of mapping one Swedish or German word into two English words. Model 3 however introduces fertility based alignment (Och & Ney, 2003), which considers such one to many translations probable.

Model 4

Adding more depth to the alignment model, relative word order is considered in Model 4. In Model 4, words are divided into word classes, and the relative word order probability is made dependant on these word classes. Model 5 which followed after Model 4 does not add more parameters to the model, but does however include modifications that improve the models efficiency.

2.2. Heuristic Alignment Models

Heuristic alignment methods differ from statistical alignment methods by being based on specific associative measures rather than pure statistical measures. Examples of such measures are the Dice coefficient, mutual information, and linguistic dictionaries.

2.2.1 Dice coefficient association

One of the most basic heuristic word alignment methods is to use the Dice-coefficient to assign co-occurrence scores to each word pair. For each segment, non-overlapping word pairs are chosen as the aligned words based on the highest co-occurrence score.

$$dice(i,j) = \frac{2 \cdot C(e_i, f_j)}{C(e_i) \cdot C(f_j)}$$

Equation 5.

The Dice co-efficient for word alignment in a sentence context is calculated by counting the number sentences where the words co-occur (C(e,f)), the number of occurrences of the english word (C(e)) and the number of occurrences of the foreign word (C(f)) and enter the values into Equation 6.

2.2.2 The K-vec alignment algorithm, 1994

The K-vec alignment algorithm (Fung & Church, 1994) is a heuristic alignment algorithm that is primarily used for dictionary extraction. It uses several heuristic measures to estimate word correspondances. The heuristics are applied in sequence, i.e. heuristic A is used to make a first candidate selection, heuristic B is used to narrow down the selection made by heuristic A and so on. The first stage alignment selection is made based on word occurrence vectors. The corpora to be analyzed is divided into K number of segments and for each word, a K-dimensional binary vector is constructed. If the word appears in segment k, the corresponding dimension in the vector is set to 1. For a document divided into three segments, where the word "car" occurs in segment 1 and 2, the binary vector for car would be <1,0,1>. All source and target language words are assigned a k-vector, and are later compared with each other. High correlation between segments where the word occurs and does not occur is thought to raise the likelihood that a certain source word corresponds with a certain target word. The selected alignments are further tested using mutual information and t-score metrics. To improve performance and results of the K-vec algorithm, the value of K must be calibrated and a threshold must be set for the lowest word frequency for a word for it to be part of a candidate word pair. The K-vec algorithm however does not describe how these selections can be made optimally.

2.2.3 Simple Hybrid Aligner, 1998

The Simple Hybrid Aligner (Ahrenberg, Andersson, & Merkel, 1998)combines the K-vec approach with a word-to-word search algorithm described by (Melamed, 1997). The Simple Hybrid Aligner tries to align words in sentences by using K-vec derived word pairs, but adds to the approach an optional weighting module that increases alignment likelihood based on the distance between the words to be aligned. The Simple Hybrid Aligner also has a phrase module for aligning multiword units (MWUs). It does so by having a multi word unit dictionary in addition to having the single word dictionary. A final improvement is the morphological module which is run post alignment in order to retrieve inflective variants of the found alignments.

2.2.4 Geometric alignment, Melamed 1999

Dan Melamed's approach (Melamed, 1999) to produce what he calls bitext maps, i.e. maps of correspondances bitexts is based on finding true points of correspondances (TPCs) in a *text-to-text-matrix*.

Melamed has named this algorithm SIMR, Smooth Injective Map Recognizer. There are two phases to the mapping algorithm, the *point generation phase*, and the *recognition phase*. During the point recognition phase, a rectangular search area is defined and SIMR generates candidate points of correspondence in this area. These points are examined during the recognition phase in hope of finding chains. If no chains can be found, the search area is expanded, and the generation-recognition cycle is repeated. SIMR works under the assumption that true bitext maps, complete bitext maps, are monotonically increasing functions.

Point generation

SIMR uses several heuristics, matching predicates, to generate candidate TPCs. The matching predicates use different kinds of information such as translation lexicons and cognate detection. The cognates used can be either orthographic or phonetic. SIMR uses the *Longest Common Subsequence Ratio* (LCSR) to score potential orthographic cognates which observes the longest sequence of matching characters, with or without gaps, between two words, and divides it with the length of the longest word-

Besides cognates, SIMR uses a *seed translation lexicon*, "a simple list of word pairs that are believed to be mutual translations". The point generation phase can also use stop lists to exclude known false friends not to be recognized as cognates.

One problem that can arise during the point generation phase is that noisy points are generated. Melamed's approach to remove such noisy points is to use a filter based on *maximum point ambiguity level*, a measure that considers the number of competing neighbors a candidate TPC has.

Recognition phase

When points have been generated and filtered, SIMR tries to identify TPC chains. By definition, chains have to be injective. Chains are also rejected if the angle of the chain deviates to much from the bitext slope (the diagonal in the bitext matrix) or, if the chain has points that are to far from the candidate chain's own least-squares line. To reduce computational complexity, SIMR uses a has a fixed chain size.

A *True Bitext Map* (TBM) is not necessarily linear even though it is injective. To find nonlinear alignment chains, SIMR uses overlapping search rectangles, which in turn generates overlapping chains. Conflicting chains are eliminated using the heuristic that chains that conflict with many other chains are more likely to be wrong. Melamed removes the most conflicting chain first, recalculates the number of conflicts, then removes the most conflicting chain again until no more conflicts are left.

Segment alignment using GSA

Segment alignments can be retrieved using information on segment boundaries in the bitext, together with the bitext map produced by SIMR. Basically, GSA uses the TPC suggestions from SIMR to find aligned segment blocks. When GSA stumbles upon a difficult case, it uses Gale and Church's character length based alignment algorithm. For the cases when GSA finds a single sandwiched segment, it assumes it to be aligned. If more then one segment is sandwiched, GSA applies Gale and Church's algorithm to the segments.

2.2.5 Clue based alignement, 2003

A word alignment technique which also uses multiple heuristic alignment strategies is the Clue alignment approach proposed by Jörg Tiedemann (Tiedemann, 2003). Tiedemanns approach however, uses a segment-to-segment matrix, a clue matrix, to represent the possible alignments, and assigns a score to each available word alignment. The score the result from a weighted summarization of the independent clue sources. Examples of such clue sources presented in (Tiedemann, 2003) are the Dice co-efficient, longest common subsequence ratio, POS tags, positional weighting, N-grams, and chunks (n-grams and chunks are used as clues for multi word unit discovery).

3. Discussion

Development in the field of automatic bitext alignment methods seems to be biased towards the statistical approach as of today. One reason might be that statistical methods scale better with larger corpora, combined with the popularity of statistical machine translation, where bitext alignment is performed to generate translation models.

4. References

Ahrenberg, L., Andersson, M., & Merkel, M. (1998). A simple hybrid aligner for generating lexical correspondences in parallel texts. *Proceedings of the 17th international conference on ….*

Fung, P. & Church, K. W. (1994). K-vec: a new approach for aligning parallel texts. *Proceedings of the 15th conference on Computational ….*

Gale, W. A. & Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*.

Melamed, I. D. (1997). A word-to-word model of translational equivalence. *Proceedings of the 35th annual meeting on Association for ….*

Melamed, I. D. (1999). Bitext maps and alignment via pattern recognition. *Computational Linguistics*.

Och, F. J. & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics,* 19-51.

Tiedemann, J. (2003). *Combining Clues for Word Alignment*. Paper presented at the EACL 2003, Department of Linguistics, Uppsala university, Sweden.