A Chunk-Driven Bootstrapping Approach to Extracting Translation Patterns

Lieve Macken[1,2] and Walter Daelemans[3]

[1] LT3, University College Ghent,
Groot-Brittanniëlaan 45, Ghent, Belgium
[2] Dept. of Applied Mathematics and Computer Science, Ghent University,
Krijgslaan 281(S9), Ghent, Belgium
[3] CLiPS Computational Linguistics Group, University of Antwerp
Prinsstraat 13, 2000 Antwerpen, Belgium

Abstract. We present a linguistically-motivated sub-sentential alignment system that extends the intersected IBM Model 4 word alignments. The alignment system is chunk-driven and requires only shallow linguistic processing tools for the source and the target languages, i.e. part-of-speech taggers and chunkers.

We conceive the sub-sentential aligner as a cascaded model consisting of two phases. In the first phase, anchor chunks are linked based on the intersected word alignments and syntactic similarity. In the second phase, we use a bootstrapping approach to extract more complex translation patterns.

The results show an overall AER reduction and competitive F-Measures in comparison to the commonly used symmetrized IBM Model 4 predictions (intersection, union and grow-diag-final) on six different text types for English-Dutch. More in particular, in comparison with the intersected word alignments, the proposed method improves recall, without sacrificing precision. Moreover, the system is able to align discontiguous chunks, which frequently occur in Dutch.

Keywords: chunk alignment, word alignment, parallel corpora, computer-aided translation.

1 Introduction

Sub-sentential alignments are used among other things to create phrase tables for statistical phrase-based machine translation systems. A stand-alone sub-sentential alignment module however, is also useful for human translators if incorporated in CAT-tools, e.g. in sub-sentential translation memory systems [1], or for bilingual terminology extraction [2,3].

In the context of statistical machine translation, GIZA++ [4] is one of the most widely used word alignment toolkits. GIZA++ implements the IBM models [5] and is used in Moses [6] to generate the initial source-to-target and target-to-source word alignments after which a symmetrization heuristic combines the

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 394–405, 2010.

alignments of both translation directions. *Intersecting* the two alignments results in an overall alignment with a high precision, while taking the *union* of the alignments results in an overall alignment with a high recall. The default symmetrization heuristic applied in Moses (*grow-diag-final*) starts from the intersection points and gradually adds alignment points of the union to link unaligned words that neighbour established alignment points. The main problem with the *union* and the *grow-diag-final* heuristics is that the gain in recall causes a substantial loss in precision, which poses a problem for applications intended for human users.

A considerable amount of research has been devoted to the topic of improving the accuracy of word and phrase alignment models. Ganchev et al. [7] also favour the idea of using intersected word alignments *by encouraging the models to agree by training them concurrently*, rather than training the alignment models in two directions and combining their predictions. Zhang et al. [8] unify the training of the word and phrase alignment models. In their staged training procedure, they first train a word alignment model and use the confident word links to reduce the phrasal alignment space. We also use a staged training procedure starting from confident word links, but in our alignment system, we use linguistic constraints to align linguistically-motivated chunks.

Several researchers demonstrated that the addition of linguistic information can improve statistically-based word alignment systems. DeNero and Klein [9] use a syntax-aware distortion component to improve the word alignments. Tiedemann [10] combines association measures with additional linguistic heuristics based on part-of-speech, phrase type, and string similarity measures. While Tiedemann makes use of chunk information, the alignment process remains word-based. In our approach, the alignment process is primarily chunk-driven.

2 Architecture

The global architecture of our system is visualized in Figure 1. The sub-sentential alignment system takes as its input sentence-aligned texts, together with additional linguistic annotations (part-of-speech codes and chunk information) for the source and the target texts along with the intersected word alignments generated by the GIZA++ toolkit. The system stores all this information in a lexical link matrix.

The sub-sentential alignment system itself is conceived as a cascaded model consisting of two phases. The objective of the first phase is to link *anchor chunks*, i.e. chunks that can be linked with a very high precision. Those anchor chunks are linked based on the intersected word alignments and syntactic similarity. In the second phase, we use a bootstrapping approach to extract language-pair specific translation rules. The anchor chunks and the word alignments of the first phase are used to limit the search space in the second phase.

Although the global architecture of our sub-sentential alignment system is language-independent, some language-specific resources are used. The system

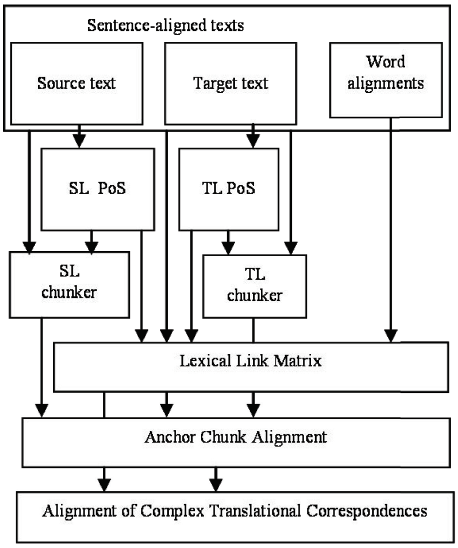

Fig. 1. Outline architecture

requires shallow linguistic processing tools for the source and the target languages, i.e. part-of-speech taggers and chunkers. We focus on the English-Dutch language pair.

2.1 Additional Linguistic Annotations

Part-of-speech tagging for English was performed by the memory-based PoS tagger MBT, which is part of the MBSP tools [11]. Part-of-speech tagging for Dutch was performed by TADPOLE [12].

We further enriched the corpora with chunk information. During text chunking, syntactically related words are combined into non-overlapping chunks based on PoS information [13]. We developed rule-based chunkers for English and Dutch. The rule-based chunkers contain distituency rules, i.e. the rules add a chunk boundary when two part-of-speech codes cannot occur in the same constituent. The following example shows an English-Dutch sentence pair divided in non-overlapping chunks:

En: It | is | a complicated | and | difficult problem
Nl: Het | is | een gecompliceerd | en | moeilijk probleem

2.2 Anchor Chunk Alignment

The basic idea behind our approach is that – at least for European languages – translations conveying the same meaning use to a certain extent the same building blocks from which this meaning is composed: i.e. we assume that to a large extent noun and prepositional phrases, verb phrases and adverbial phrases in one language directly map to similar constituents in the other language. The extent to which our basic assumption holds depends on the translation strategy that was used. Text types that are typically translated in a more literal way (e.g. technical texts) will contain more direct correspondences than text types for which a more free translation strategy was adopted (e.g. journalistic texts).

In the first phase of this system, *anchor chunks* are linked, i.e. chunks that can be linked with a very high precision. Chunks are considered to be anchor chunks if all words of source and target chunk(s) are either linked by means of a lexical link or can be linked on the basis of corresponding part-of-speech codes.

2.3 Alignment of Complex Translational Correspondences

In the second phase, we use a bootstrapping approach to align more complex translational correspondences. We start from a sentence-aligned parallel corpus of short sentences in which anchor chunks have been aligned on the basis of the intersected GIZA++ alignments and syntactic similarity.

The bootstrapping process is a cyclic process which alternates between extracting candidate translation rules (extraction step) and scoring and filtering the extracted candidate translation rules (validation step). From the second bootstrapping cycle onwards, the validated translation rules are first applied to the corpus, after which the extraction process is launched again. The bootstrapping process is repeated four times.

2.4 Extraction Step

In the extraction step, candidate translation rules are extracted from unlinked source and target chunks. Different alignment types (1:1, 1:n, n:1 and n:m) are considered:

- From sentence pairs that only contain 1:1, 1:n and n:1 unlinked chunks, candidate translation rules that link 1:1, 1:n and n:1 chunks are extracted. In the left example of Figure 2, the source chunk *membership* and target chunk *het lidmaatschap* are selected because they are the only unlinked chunks in the sentence pair.
- From sentence pairs in which the only unlinked chunks in the source or target sentence are lexically interlinked, candidate translation rules that link n:m chunks are extracted. In the right example of Figure 2, the source chunks *not just* | *an old person's disease* and the target chunks *geen ziekte* | *die* | *alleen ouderen* |*treft* [En: *no disease that just elderly strikes*] are selected, as the source chunks *not just* and *an old person's disease* are lexically interlinked and are the only unlinked chunks in the source sentence.

From the selected source and target chunks two types of rules are extracted: *abstract* rules and *lexicalized* rules. The rules can be contiguous or non-contiguous.

- Abstract rules are coded as PoS sequences. Established word alignments within the extracted chunks are coded as indices. For example the rule DET+N-gen+N_1 → DET+N_1|PREP|DET+N captures the transformation of a genitive into a prepositional phrase as in *the public's right* → *het recht van de burgers* [En: *the right of the public*]
- Lexicalized rules are coded as token sequences, e.g. *to treat* → *ter behandeling van* [En: *for the treatment of*]

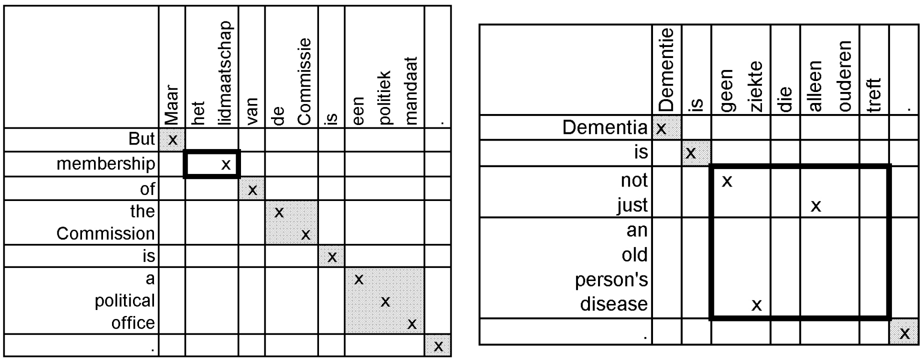

Fig. 2. Sentence pair with one unlinked source (*membership*) and target chunk (*het lidmaatschap [En: the membership]*) and sentence pair with unlinked source chunks that are grouped by means of lexical links. Chunk boundaries are indicated by horizontal and vertical lines, intersected IBM Model 4 word alignments by x's, and anchor chunks in light grey.

From the second bootstrapping cycle onwards, the validated rules are first applied to the whole training corpus, resulting in new translation pairs containing 1:1, 1:n and n:1 unlinked chunks, after which the extraction process is launched again.

The matching process considers all lexically interlinked groups of chunks (see the right example of Figure 2) and all unlinked source and target chunks with a neighbouring left or right anchor chunk and uses the word aligments and the anchor chunks to build up the target or source pattern.

2.5 Validation Step

The aim of the validation step is twofold. On the one hand, we want to extract a subset of *reliable* translation rules out of the set of candidate translation rules. On the other hand, we want to sort the translation rules to determine the order in which the rules are applied.

We use the Log-Likelihood Ratio as statistical association measure to compute an association score between each source and target pattern of all candidate translation rules. The Log-Likelihood ratio has been used before for building translation dictionaries [14] and for word alignment [15].

To compute the Log-Likelihood Ratio, we first count for each candidate translation rule how many times the source and target pattern co-occur in the corpus and calculate the Log-Likelihood on the basis of the observed frequencies and the expected frequencies under the null hypothesis of independence as follows:

$$-2log(\lambda) = 2\sum_{ij} O_{ij}log(\frac{O_{ij}}{E_{ij}}). \quad (1)$$

Dunning [16] showed that the Log-Likelihood ratio test allows comparisons to be made between the significance of the occurrences of both rare and common phenomena, which makes the test appropriate for our purposes. According to Manning and Schütze [17], $-2log(\lambda)$ has a distribution similar to that of chi-square can thus be used for hypothesis testing using the statistical tables for the distribution of chi-square. For a contingency table with two rows and two columns the critical value is 10.83 for the significance level of 0.001 [18].

Therefore, in the validation step, we only retain translation rules with a Log-Likelihood value higher than 10.8. To reduce the memory requirements of our system, we only validated candidate translation rules that co-occurred at least 5 times.

3 Experimental Results

3.1 Bootstrapping Corpus

For the extraction and the validation step of the bootstrapping process we extracted two subcorpora from the Dutch Parallel Corpus [19].

- The first subcorpus contains 36,406 sentence pairs (478,002 words) of short sentences (1-10 words).
- The second subcorpus contains 79,814 sentence pairs (1,892,233 words) of medium-length sentences (1-20 words).

The Dutch Parallel Corpus has a balanced composition and contains five text types: administrative texts, texts treating external communication, literary texts, journalistic texts and instructive texts. All text types are present in the selected subcorpora.

3.2 Reference Corpus

In order to evaluate the bootstrapping approach, a manual reference corpus was created that includes six different text types: journalistic texts, proceedings of plenary debates (selected from Europarl), financial newsletters, press releases, technical texts of the medical domain, and user manuals[1]. The formal characteristics of the reference corpus are presented in Table 1.

[1] The largest part of the manual reference corpus is publicly available as part of the Dutch Parallel Corpus, which is distributed by the Agency for Human Language Technologies (TST-centrale).

We assume that for each of the three text types another translation style was adopted, with the journalistic texts being the most free and the technical texts and user manuals being the most literal translations.

In the manual reference corpus, different units could be linked (words, word groups, paraphrased sections, punctuation). Three different types of links were used: regular links for straightforward correspondences, fuzzy links for translation-specific shifts of various kinds, and null links for words for which no correspondence could be indicated (deletions or additions).

To evaluate the system's performance, the links created by the system were compared with the links of the manual reference files.

Table 1. En-Nl Test data

Text type	# Words	# Sentences	# Texts
Journalistic texts	8,557	177	3
Proceedings EP	3,139	105	7
Newsletters	12,000	344	2
Press Releases	4,926	212	4
Technical texts	8,661	216	4
User Manuals	4,010	296	2
Total	41,293	1,350	22

To be able to compare the alignments of the system with the reference alignments, all phrase-to-phrase alignments were converted into word-to-word alignments by linking each word of the source phrase to each word of the target phrase (all-pairs heuristic).

3.3 Evaluation Metrics

The evaluation of word alignment systems is not a trivial task. Different evaluation metrics exist, and they mainly differ in the way divergent translational correspondences are treated. Given the controversy in the literature, we evaluated our system with two different metrics: Alignment Error Rate (AER) and a weighted version of F-Measure.

Alignment Error Rate. Alignment error rate was introduced by Och and Ney [4] to evaluate the performance of word alignment systems. They distinguished *sure* alignments (S) and *possible* alignments (P) and introduced the following redefined precision and recall measures (where A refers to the set of alignments):

$$precision = \frac{|A \cap P|}{|A|}, recall = \frac{|A \cap S|}{|S|}. \tag{2}$$

and the alignment error rate (AER):

$$AER(S, P; A) = 1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}. \tag{3}$$

The distinction between *sure* and *possible* alignments approximately corresponds to the distinction between regular and fuzzy links in our annotation scheme. Therefore we consider all regular links of the manual reference as *sure* alignments and all fuzzy and null links as *possible* alignments to compare the output of our system with the manual reference.

Weighted F-Measure. F-Measure combines the traditional precision and recall metrics and can be calculated on all word-to-word links. However, Melamed [20] pointed out that F-Measure poses a problem. If precision and recall is calculated on all word-to-word links, all links would be equally important and would place undue importance on words that were linked more than once (e.g. all word-to-word links resulting from the phrasal alignments). Therefore, Melamed introduced a weighted version of precision and recall in which a weight is assigned to each word-to-word link.

We use the weighting method developed by Davis [21], which is a refinement of Melamed's weighting principles. In this weighting scheme, every word contributes 0.5 to the total weight. In case of interlinked word-to-word links from the phrasal alignments, each link is assigned the total weight of the phrasal alignment divided by the number of word-to-word links. Precision and recall are then calculated on the normalized weights.

3.4 Results

The results of all our experiments are summarized in Tables 2 and 3. In Table 2, we give per text type the alignment scores for the symmetrized IBM Model 4 predictions, using the three most commonly used symmetrization heuristics: intersection (∩), union (∪), and grow-diag-final (Gdf). As expected, the intersection heuristic generates the most precise overall alignment, while the union results in an alignment with the highest recall. The recall gain in the *union* and *grow-diag-final* heuristics causes a substantial loss in precision.

In Table 3, the results of our chunk-based extension to the intersected IBM Model 4 alignments are given for four different settings:

- **10Lex:** bootstrapping corpus of short sentences (1-10 words); only lexicalized translation rules or abstract rules containing lexical indices are retained in the validation step
- **10All:** bootstrapping corpus of short sentences (1-10 words), all lexicalized translation rules or abstract rules containing lexical indices are applied first; in a second step abstract rules without lexical clues are applied
- **20Lex:** identical to 10Lex but a bootstrapping corpus of medium-length sentences (1-20 words) is used
- **20All:** identical to 10All but a bootstrapping corpus of medium-length sentences (1-20 words) is used

The results reflect the different translation strategies of the different text types: the technical texts are the easiest to align; the journalistic and Europarl texts

Table 2. Results for the different symmetrized IBM Model 4 predictions: intersection (∩), union (∪), and grow-diag-final (Gdf) expressed in terms of AER and weighted F-measure

	Journalistic			Europarl			Newsletters		
	∩	∪	GDF	∩	∪	GDF	∩	∪	GDF
Prec	95.7	57.9	62.0	94.1	73.7	76.1	96.4	72.3	76.3
Rec	65.5	84.2	83.7	64.0	80.0	79.1	65.4	84.7	83.9
AER	21.8	32.4	29.6	22.9	23.6	22.6	21.8	22.3	20.3
WPrec	95.7	58.8	62.9	94.1	75.7	77.9	96.4	72.5	76.5
WRec	51.5	67.6	67.0	51.1	64.9	63.9	58.9	75.5	74.7
WF1	67.0	62.9	64.9	66.2	69.9	**70.2**	73.1	74.0	75.6
	Pressreleases			Technical			User Manuals		
Prec	98.6	76.2	80.7	97.8	78.0	81.3	97.8	73.2	77.8
Rec	63.3	76.3	75.5	73.2	88.0	87.4	64.1	83.4	82.5
AER	22.7	23.8	21.9	16.1	17.5	15.9	22.3	22.3	20.0
WPrec	98.6	77.3	81.7	97.8	78.5	81.8	97.8	74.1	78.7
WRec	64.4	76.3	75.8	68.1	80.9	80.3	61.0	78.1	77.5
WF1	77.9	76.8	78.6	80.3	79.7	81.1	75.1	76.1	78.1

Table 3. Results for the chunk-based extension to the intersected IBM Model 4 alignments for four different settings expressed in terms of AER and weighted F-measure

	Journalistic				Europarl				Newsletters			
	10Lex	10All	20Lex	20All	10Lex	10All	20Lex	20All	10Lex	10All	20Lex	20All
Prec	94.0	93.0	92.4	92.0	93.8	92.4	93.4	92.0	96.0	94.9	94.6	94.6
Rec	70.1	71.9	70.6	73.1	67.8	68.4	68.7	69.2	69.8	71.8	70.2	72.4
AER	19.3	18.6	19.7	**18.2**	20.4	20.5	**19.9**	20.1	8.9	18.0	19.1	**17.7**
WPrec	94.1	93.1	93.3	92.1	93.9	92.8	93.5	92.4	95.9	94.9	95.8	94.6
WRec	55.0	56.6	55.7	57.6	54.1	54.8	54.8	55.4	62.4	64.3	63.1	64.8
WF1	69.4	70.4	69.8	**70.9**	68.6	68.9	69.1	69.3	75.7	76.7	76.1	**76.9**
	Pressreleases				Technical				User Manuals			
Prec	98.2	97.6	97.7	96.8	97.2	96.4	96.1	96.3	96.6	96.4	96.3	95.7
Rec	65.3	66.4	65.8	66.9	76.3	77.7	77.3	78.4	68.0	69.9	68.8	70.8
AER	21.3	20.7	21.1	**20.6**	14.3	13.7	14.1	**13.3**	19.8	18.6	19.4	**18.3**
WPrec	98.3	97.8	97.7	97.0	97.3	96.6	96.9	96.4	96.7	96.5	96.4	95.8
WRec	66.5	67.6	67.0	68.1	70.9	72.3	72.1	73.0	65.0	66.6	65.7	67.2
WF1	79.3	79.9	79.5	**80.1**	82.0	82.7	82.6	**83.1**	77.7	78.8	78.1	**79.0**

the most difficult. In all settings, the results show an overall AER reduction over all symmetrized IBM Model 4 predictions. In terms of weighted F-Measure, the results show a higher F-score for all text types except for the Europarl texts.

For all text types and in all experimental settings, the proposed system improves the recall of the intersected IBM Model 4 word alignments without sacrificing precision.

Overall, enlarging the training set has a positive effect on the system's performance. More precise results can be obtained by only allowing translation rules that contain lexical clues (either abstract PoS rules with lexical indices or lexicalized rules).

Table 4 gives an overview of the total number of validated rules in the different experimental settings and gives details on the number of discontiguous (either abstract or lexicalized) and lexicalized validated rules. As expected, the number of validated rules increases if the corpus size is increased. If only translation rules that contain lexical clues are allowed, the number of validated translation rules is drastically reduced. The share of discontiguous rules ranges from 23 to 39%; the share of lexicalized rules from 31 to 48%.

On the right-handside of the table, the number of applied rules in the different test corpora is given. In order to process the 40,000 words of the test corpora, 14 to 24% of the rules are applied. The share of discontiguous rules accounts for 14 to 20%.

Some example rules are given below:

- N_1 → DET+N_1 (*History → de Geschiedenis*)
- DET_1+N_2+N → DET_1+N_2 (*a movie producer → een filmproducent*)
- PREP_1+V-prpa_2 → PREP_1 | DET+N_2 | PREP (*for managing → voor het management van*)
- DET_1+N_2;PREP | N_3 → DET_1+N_2+N_3 (*a number of events → een aantal evenementen*)
- V-fin_1+V-papa_2 → V-fin_1 ... V-papa_2 (*had written → had ... geschreven*)
- ADV_1;...;ADJ_2 → ADV_1+ADJ_2 (*not;...;longer → niet langer*)
- last → voor het laatst
- has → beschikt ... over
- agree → ben | het ... eens

The most frequently applied rules take care of the insertion of a determiner in Dutch (e.g. *History → de Geschiedenis*) or deal with Dutch compounds of which only a part has been aligned by the GIZA++ intersected word alignments (e.g. *filmproducent*). The most frequently applied discontiguous rules deal with verbal groups that are often split in Dutch (*had ... geschreven*). However, other discontiguous chunks are captured as well. The discontiguous lexicalized rules are able to deal with phrasal verbs (e.g. *beschikken ... over*).

Table 4. Total number of validated rules in the different settings and number of validated discontiguous and lexicalized rules; total number of applied rules in the test corpora and number of applied discontiguous and lexicalized rules

	Validated			Applied		
	Total	Discont.	Lexicalized	Total	Discont.	Lexicalized
10Lex	1526	344	724	303	46	70
10All	2135	574	744	508	104	70
20Lex	3790	1174	1828	530	108	153
20All	5826	2287	1828	872	249	153

4 Conclusion and Future Work

We developed a new chunk-based method to add language-pair specific knowledge – derived from shallow linguistic processing tools – to statistical word alignment models. The system is conceived as a cascaded model consisting of two phases. In the first phase *anchor chunks* are linked on the basis of the intersected IBM Model 4 word alignment and syntactic similarity. In the second phase, we use a bootstrapping approach to extract language-pair specific translation patterns.

We demonstrated that the proposed system improves the recall of the intersected IBM Model 4 word alignments without sacrificing precision, which makes the resulting alignments more useful for incorporation in CAT-tools or bilingual terminology extraction tools. Moreover, the system's ability to align discontiguous chunks makes the system useful for languages containing split verbal constructions and phrasal verbs.

As the chunk-based extension aligns chunks rather than words, we assume that incorporation of these precise chunks in the SMT phrase tables has a positive impact on Machine Translation quality as well. In future work, we would like to evaluate our approach in an existing phrase-based SMT system.

References

1. Planas, E.: SIMILIS Second-generation translation memory software. In: 27th International Conference on Translating and the Computer (TC27), London, United Kingdom, ASLIB (2005)
2. Itagaki, M., Aikawa, T., He, X.: Automatic Validation of Terminology Consistency with Statistical Method. In: Machine Translation Summit XI. European Associaton for Machine Translation, pp. 269–274 (2007)
3. Macken, L., Lefever, E., Hoste, V.: Linguistically-based Sub-sentential Alignment for Terminology Extraction from a Bilingual Automotive Corpus. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, United Kingdom (2008)
4. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
5. Brown, P.F., Della Pietra, V.J., Della Pietra, S.A., Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19(2), 263–311 (1993)
6. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Czech Republic, Prague. Association for Computational Linguistics, pp. 177–180 (2007)
7. Ganchev, K., Graça, J.V., Taskar, B.: Better Alignments = Better Translations? In: Proceedings of ACL 2008: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 986–993 (2008)

8. Zhang, H., Quirk, C., Moore, R.C., Gildea, D.: Bayesian Learning of Non-Compositional Phrases with Synchronous Parsing. In: Proceedings of ACL 2008: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 97–105 (2008)
9. DeNero, J., Klein, D.: Tailoring Word Alignments to Syntactic Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic. Association for Computational Linguistics, pp. 17–24 (2007)
10. Tiedemann, J.: Combining Clues for Word Alignment. In: Proceedings of the 10th Conference of the European Chapter of the ACL (EACL 2003), Budapest, Hungary (2003)
11. Daelemans, W., van den Bosch, A.: Memory-based language processing. Cambridge University Press, Cambridge (2005)
12. van den Bosch, A., Busser, B., Daelemans, W., Canisius, S.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting, Leuven, Belgium, pp. 191–206 (2007)
13. Abney, S.: Parsing by chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) Principle-Based Parsing. Kluwer Academic Publisher, Dordrecht (1991)
14. Melamed, D.I.: Models of translational equivalence among words. Computational Linguistics 26(2), 221–249 (2000)
15. Moore, R.C.: Association-Based Bilingual Word Alignment. In: ACL Workshop on Building and Using Parallel Texts, Ann Arbor, Michigan, United States, pp. 1–8 (2005)
16. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)
17. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. Massachusetts Institute of Technology (2003)
18. McEnery, T., Xiao, R., Yukio, T.: Corpus-based Language Studies. An advanced resource book. Routledge, London (2006)
19. Macken, L., Trushkina, J., Rura, L.: Dutch Parallel Corpus: MT corpus and Translator's Aid. In: Machine Translation Summit XI, Copenhagen, Denmark, pp. 313–320 (2007)
20. Melamed, D.I.: Empirical Methods for Exploiting Parallel Texts. MIT Press, Cambridge (2001)
21. Davis, P.C.: Stone Soup Translation: The Linked Automata Model, Unpublished PhD, Ohio State University (2002)