Improving Lexical Alignment Using Hybrid Discriminative and Post-Processing Techniques

Paulo Schreiner[1], Aline Villavicencio [1], Leonardo Zilio [2], Helena M. Caseli [3]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]Instituto de Letras – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[3]Departamento de Computação – Universidade Federal de São Carlaos (UFSCar)
Rodovia Washington Luís, km 235 – 13.565-905 – São Carlos - SP - Brazil

{pschreiner,avillavicencio}@inf.ufrgs.br
leonardozilio@yahoo.de, helenacaseli@dc.ufscar.br

Abstract. *Automatic lexical alignment is a vital step for empirical machine translation, and although good results can be obtained with existent models (e.g. Giza++), more precise alignment is still needed for successfully handling complex constructions such as multiword expressions. In this paper we propose an approach for lexical alignment combining statistical and linguistic information. We describe the development of a baseline discriminative aligner and a set of language dependent post-processing functions that allow the inclusion of shallow linguistic knowledge. The post-processing functions were designed to significantly improve word alignment mainly on verb-particle constructs both over our baseline and over Giza++.*

1. Introduction

Automatic lexical (word) alignment is a previous step necessary for the creation of an empirical machine translation system [Brown et al. 1990, Somers 1999]. Given a sentence-aligned parallel corpus, the lexical alignment process consists of producing a set of relations (alignments) between the lexical units of both languages. The lexical units are usually simple words, but may also be multiword expressions (MWEs), which can be defined as "idiosyncratic interpretations that cross word boundaries (or spaces)" [Sag et al. 2002].

The task of lexical alignment has traditionally been done in a completely statistical, unsupervised manner, whereby a generative aligner tries to infer the parameters of a model of the statistical process by which a source sentence generates a target one. [Brown et al. 1993] describe a set of increasingly complex generative processes (the IBM models) and algorithms to estimate its parameters.

The difficulty in adding complex linguistic knowledge to generative models has given rise to several so-called discriminative lexical aligners such as in [Moore 2005] and in [Niehues and Vogel 2008]. In a discriminative aligner it is not necessary to model a complex statistical problem, instead a set of feature functions are created each one capturing a specific facet of a word alignment. Thus, the features are combined (usually linearly) and it is then necessary to search the space of all possible alignments

Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology, pages 97–106,
Cuiabá, MT, Brazil, October 24–26, 2011. ©2011 Sociedade Brasileira de Computação

for the one which has the highest score. For example, [Moore 2005] linearly combines a small number of features, while [Liu et al. 2005] propose a very similar technique, but using a log-linear combination of the individual features. With respect to parameter training, techniques range from voted perceptron [Moore 2005] to conditional random field [Niehues and Vogel 2008], while [Fraser and Marcu 2006] go even further and propose a semi-supervised parameter optimization for their discriminative aligner. The search for the best alignment is made with a modified hillclimbing method in [Fraser and Marcu 2006], while [Liu et al. 2005] use a greedy algorithm.

One subject which has drawn comparatively little attention is the correct alignment of MWEs. MWEs are a significant part of the lexicon of a speaker, perhaps as numerous as the single words [Jackendoff 2002], and various techniques to identify and process them have been proposed using different kinds of information, from syntax [Baldwin 2005] to statistics [Ramisch et al. 2008, Evert and Krenn 2005]. In this work we focus on a specific type of MWEs, namely verb-particle constructs (VPCs), which are combinations of verb and particle such as *turn up* and *made up*. In terms of syntax, VPCs can have complex subcategorization frames, such as transitive VPCs, which take a NP argument between the verb and the particle, e.g.

He made <u>the whole story</u> up.

The semantics of VPCs ranges from more transparent (e.g. *clear up* where the particle introduces a sense of completion) to more opaque cases (e.g. *make out* as kiss).

An adequate processing of MWEs is important for precise machine translation and can benefit from being taken into account during the task of lexical alignment [Deksne et al. 2008]. Therefore, in this paper we apply machine learning techniques to train post-processing heuristics that can be used to enrich a baseline alignment. We show how this approach can be applied to identify VPCs in the task of lexical alignment and, by providing a more precise alignment of VPCs, improve the general performance over the baseline.

This paper is structured as follows: in section 2 we describe the resources that were used and in section 3 the alignment methods developed. Section 4 discusses the results obtained, and finally in section 5 we present our conclusions and suggest directions to future research.

2. Resources

The parallel corpus used in our experiments was the Opus subtitle corpus [Tiedemann 2009]. This corpus was built using freely available movie subtitles found in the Web, which were automatically sentence aligned.[1] For the experiments reported in this paper, we used the English-Portuguese portion of the Opus corpus after some preprocessing steps to remove tags and to tokenize words. Table 1a presents the total amount of sentences and tokens in each language, where *en* stands for English and *pt* for Portuguese.

The parameter estimation of Giza++ was performed based on the complete unannotated Opus corpus [Och and Ney 2003], since it is an unsupervised process. A manually annotated corpus was also needed to both estimate parameters of our discriminative

[1] The automatic sentence alignment was not manually corrected.

Lang	# Sentences	# Tokens
en	351,106	3,077,113
pt	351,106	2,605,376

(a) The opus en-pt subtitle corpus

Corpus	# Sent	# Alignments	# VPC
Complete tune	600	4019	78
Partial tune	900	206	103
Complete test	500	4395	107
Partial test	600	142	71

(b) The corpora use in the exeriments

Table 1. Corpora

aligner and also to evaluate it. This gold standard corpus was built from a selected a subset of the Opus corpus following two types of annotation that gave rise to two different corpora:

- the **complete** corpus contains annotations for all the correspondences (pairs of words or MWEs alignments)
- the **partial** corpus contains annotations only for the correspondences involving VPCs

This two-fold annotation approach allows us to have a good number of VPC alignments, without having to annotate a prohibitively large portion of the corpus. VPCs in the lexical alignments were explicitly marked, both in the complete and in the partial corpus. The annotation was made using YAWAT [Germann 2008] by 2 Portuguese native speakers fluent in English, following a set of guidelines based on [Caseli et al. 2005]. Any verb in English followed by a particle or subordinating conjunction which had to be aligned as a unit was considered a VPC. In order to measure our inter-annotator agreement, we calculated the kappa (κ) measure [Carletta 1996], and obtained a value of 0.78. According to [Carletta 1996], among other authors, a value of κ between 0.67 and 0.80 indicates a good agreement between annotators.

These two corpora were further divided into **tuning** and **test** sets. The tuning set of alignments was used during the development and parameter tuning of the aligner, while the held out test set was used in the evaluation. These four different annotated corpora are summarized in table 1b. For the partial corpora, the number of sentences is not equal to the number of sentences actually annotated, because only those containing VPCs were annotated.[2] The corpora were also annotated with part-of-speech information using the Treetagger [Schmid 1994] and training data for English and Portuguese.[3]

3. Aligner

The new automatic lexical aligner proposed in this paper has a discriminative [Moore 2005] language-independent core and a small number of language-dependent post-processing heuristics. In a discriminative aligner, a series of feature functions are combined forming a global alignment score where the task is to find the alignment which

[2]Each corpus was taken from a different part of the Opus subtitle corpus, i.e., from different movies, so that we avoid over-tuning the aligner to a specific portion of the corpus.

[3]The training data for English and Portuguese is available at http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

gives the maximum global alignment score. The great advantage of discriminative alignment in relation to a generative one is that it makes it easier to incorporate new features [Fraser and Marcu 2006].

The proposed alignment process is divided into 3 phases: *search and combination* (section 3.1), *optimization* (section 3.2) and *post-processing* (sections 3.4, 3.5).

3.1. Search and combination

In the proposed discriminative lexical aligner, the search for the best alignment is made using a simple hillclimbing algorithm.[4] Therefore, starting from a NULL alignment (where there are no correspondences between words) the algorithm searches the neighborhood of the current alignment for an alignment with a better combined score. If one is found, it is set as the new current alignment and the process is repeated. If not, the algorithm terminates.

The *neighborhood* is defined as the set of alignments that can be reached through one of the following basic operations. Given that x corresponds to a word position in the source sentence and y and z correspond to word positions in the target sentence, we have four possible operations:

- **add link**, where a new individual alignment is added;
- **remove link**, where an existing individual alignment is removed;
- **move row**, where an existing alignment is moved along the row of the alignment matrix. For instance, if we have an alignment (x, y) one row movement would be to remove the alignment (x, y) and create an alignment (x, z) for all possible zs where no alignment currently exists; and
- **move column**, which is similar to the move row operation but in this case we keep the target word (y) fixed instead of the source word.

Taking into account the neighborhood of a input pair of aligned sentences, the different features of the aligner are combined linearly:

$$\text{score}_{\text{global}} = \sum_i w_i * \text{score}_i \tag{1}$$

where score_i is the score of feature i for the sentence being processed and w_i is the weight of the feature i. The features considered in our experiments are presented in section 3.3.

3.2. Parameter Optimization

The weights w_i are optimized directly with respect to the f-score, using the algorithm:

for all $currentFeatureVector_i$ **do**
  $currentFeatureVector_i \leftarrow 0.5$
end for
$step\_size \leftarrow 0.05$[5];
repeat
  for all $currentFeatureVecture_i$ **do**

[4]Even if hillclimbing algorithms may sometimes suffer from local maxima, they in general have good performance [Brown et al. 1993].

[5]This $step\_size$ was empirically defined based on manual experimentation on the tuning corpus.

$featureVectorCandidates \leftarrow featureVectorCandidates \cup$ { adding and subtracting step_size from $currentFeatureVecture_i$}[6]

end for

$currentFeatureVecture \leftarrow max featureVectorCandidates$

if $currentFeatureVecture$ not changed **then**

$step\_size \leftarrow step\_size/2$

end if

until $step_size < 0.01$

Following this process, the alignment method iteratively updates the weight vector in a way that maximizes the f-score, and when no improvement is possible, it reduces the step size to refine the search. The assumption that the f-score varies smoothly and monotonically with variations in each feature weight is a simplification, consequently there is no guarantee that the algorithm will find the best weight vector. Nevertheless good results were obtained using a reasonably sized gold standard and empirically defined initial values.

3.3. Features

The word aligner uses three feature functions, namely:

1. **word translation score:** this feature is the probability that a word in English is translated to a specific word in Portuguese. This feature was calculated running Giza++ in both directions (en → pt and pt → en), and taking the average of the translation probabilities found by Giza++.
2. **fertility:** this is a measure of the probability that a word in English (Portuguese) is translated to 0, 1, 2, 3,... words in Portuguese (English). For example, the token *to* in English can be omitted in the translation to Portuguese ($fertility = 0$), or it might be translated to a single word ($fertility = 1$) or to an expression with 2 or more words ($fertility = 2, 3, \ldots$). Each of these cases has a probability, which is captured by the fertility probability feature. This feature was also taken from Giza++.
3. **coherence:** this measure assumes that words in a sentence appear in the translation roughly in the same position, and if the position changes it is usually by a small amount, or a whole *block* of words is moved. The tuning corpus was used to estimate the coherence score based on the near neighbors of each alignment.

As described so far, the algorithm is language independent, and this version is referred to in the text as **hill base**.

3.4. Heuristics

An empirical analysis of the performance of the hill base aligner has indicated problems with specific types of alignment, particularly with more complex (n:m) alignments such as VPCs. Therefore, to improve the alignment further, a series of mostly language dependent heuristics were added to the language independent core. Due to performance issues, instead of adding the heuristics as new features of the discriminative aligner, these were implemented as a post-processing step, whereby the individual alignments are updated

[6] In this way, the number of candidate feature vectors is 2^n, with n being the number of features.

through an one-pass function. By performing a post-processing step, we easily trained machine learning classifiers that can add or remove individual alignments and, in the same time, avoided the addition of too many features to the core, which would have to be executed on each step of the hillclimbing process. The proposed heuristics are:

- **intersection:** adds any individual alignment found on the intersection of the alignment of Giza++ in both directions.
- **Pron + V ⇔ V:** aligns personal pronouns (Pron) followed by verbs (V) in English to verbs in Portuguese, since in Portuguese this kind of pronoun before the verb is optional and often omitted and the information provided by it is encoded in the verb inflection (e.g. *I needed* ⇔ *precisei*, or *We realized* ⇔ *percebemos*).
  To perform this heuristic, a tree based classifier (using J48 [Zhao and Zhang 2008] implemented in Weka [Hall et al. 2009]) was trained using all the sequences Pron + V and various relevant features collected from the tuning corpus.
- **punctuation:** aligns punctuation marks using a tree based classifier, since some of the most frequent errors made by the aligner involved punctuation, which can be partly explained by the fact that translations in the corpus often omit or add punctuation. The relevant features used are the coherence score as described in section 3.3 and a series of boolean features for the immediate neighbors of the punctuation mark in question, with the idea that if they are aligned, the probability that the punctuation mark should also be aligned is higher.

We refer to the alignment resulting from the post-processing phase as **hill post**.

3.5. VPC processing

To verify the impact of adding language dependent information about complex constructions, in this work we focus on the alignment of verb-particle constructs. In order to do that, from the tuning corpus all the verb-particle candidates and the following set of features were collected:

- **verb aligned**: it is *true* if the verb in English was aligned to the verb in Portuguese by the hill post aligner, otherwise it is *false*.
- **particle aligned**: it is *true* if the particle in English was aligned to the verb in Portuguese by the hill post aligner, and otherwise it is *false*.
- **verb giza src→trg**, **verb giza trg→src**: where *trg* and *src* are the target (Portuguese) and the source (English) respectively. This feature indicates the same as *verb aligned*, but for the Giza++ English → Portuguese and Portuguese → English alignments respectively.
- **particle giza src→trg**, **particle giza trg→src**: this feature implements the same idea as *particle aligned*, but for the Giza++ English → Portuguese and Portuguese → English alignments respectively.
- **verb score**: this feature is the *word translation score* for the verb in English with the verb in Portuguese, as explained in section 3.
- **particle score**: this feature is the *word translation score* of the particle in English with the verb in Portuguese.
- **verb other**: this feature states for the ***maximum*** *word alignment probability* of the verb in English with any word other than the verb in Portuguese.

- **particle other**: this feature is the ***maximum*** *word alignment probability* of the particle in English with any word other than the verb in Portuguese.

Finally, to deal with the ambiguous TreeTagger part-of-speech tags for particles (RP), subordinating conjunctions (IN) and adverbs (RB), we searched the alignments generated by the hill post aligner using two heuristics.

In the first one, **hill VPC 1**, we search for any alignment of an English and a Portuguese verb, and selected those in which a word in the next 3 positions to the right of the verb was tagged as either RP, RB or IN. A decision tree classifier (the J48 algorithm implemented in Weka) was trained to determine whether a given VPC candidate with a set of features is a genuine VPC or not. This classifier was added as a post-processing heuristic to the proposed aligner. It should be noted that , *given* an individual alignment containing a verb in English, this heuristic finds a nearby particle and decides if it should be added to (or removed from) the verb forming a VPC (classifier $\rightarrow$ true), or not (classifier $\rightarrow$ false). If the verb is not aligned in the first place, nothing is done.

In the second heuristic, **hill VPC 2**, we do not require the verbs in the two languages to be aligned. Instead we search for verbs followed by a word tagged as RP, RB or IN in one of the next 3 positions, and generate a VPC candidate with the verb, the particle and **each** of the verbs found in the Portuguese sentence. If the classifier outputs *true*, we set the alignments of both the verb and the particle in English to the verb in Portuguese, if it outputs *false* we remove the alignment of the particle to the verb in Portuguese. Both heuristics are used to train a classifier which is added as a post-processing step.

4. Results

In this section we report the experiments carried out to evaluate the proposed alignment methods, the hill climbing aligners described in section 3, and the Giza++ aligner, using the refined symmetrization heuristic described in [Och and Ney 2003]. The evaluation was performed on previously unseen test corpora and the results are shown in tables 2a and 2b for complete and partial test corpora, respectively.

Aligner	**Precision**	**Recall**	**F-score**
Hill Base	0.736	0.612	0.668
Hill Post	0.736	0.656	0.694
Hill VPC 1	**0.737**	0.658	**0.695**
Hill VPC 2	0.734	0.657	0.694
Giza++	0.643	**0.678**	0.660

(a) Complete test corpus

Aligner	**Precision**	**Recall**	**F-score**
Hill Base	0.000	0.000	0.000
Hill Post	**1.000**	0.010	0.019
Hill VPC 1	0.946	0.340	0.500
Hill VPC 2	0.676	**0.447**	**0.538**
Giza++	0.548	0.330	0.412

(b) Partial test corpus

Table 2. Results

In order to evaluate the different models proposed we follow the *bootstraping* methodology proposed by [Zhang et al. 2004] to compare our aligners with a confidence interval of 95%, using 1000 random re-samples. First, we compared the aligners on the complete test corpus. In table 3a $\ll$ means that the aligner x is significantly worse than aligner y with respect to f-score. Similarly, $\gg$ means that aligner x is significantly better, while $\equiv$ means that no statistically significant difference on the f-score of aligner x and y, with a confidence interval of 95%, was found. The same process was repeated with the partial annotation test corpus, in order to evaluate the performance of each method regarding exclusively the VPCs. The results are shown in table 3b.

x \ y	Base	Post	VPC1	VPC2	Giza
Base	-	$\ll$	$\ll$	$\ll$	$\equiv$
Post	$\gg$	-	$\ll$	$\equiv$	$\gg$
VPC1	$\gg$	$\gg$	-	$\gg$	$\gg$
VPC2	$\gg$	$\equiv$	$\ll$	-	$\gg$
Giza	$\equiv$	$\ll$	$\ll$	$\ll$	-

(a) Complete significance

x \ y	Base	Post	VPC1	VPC2	Giza
Base	-	$\equiv$	$\ll$	$\ll$	$\ll$
Post	$\equiv$	-	$\ll$	$\ll$	$\ll$
VPC1	$\gg$	$\gg$	-	$\equiv$	$\equiv$
VPC2	$\gg$	$\gg$	$\equiv$	-	$\gg$
Giza	$\gg$	$\gg$	$\equiv$	$\ll$	-

(b) VPC Significance

Table 3. Statistical Significance

With these tests, we can conclude that all Hill X aligners are superior to both our *baseline* and Giza++ with respect to f-score and a confidence interval of 95% (with an $\alpha = 0.5$ for the general case, that is, taking into account all types of alignment), as can be noticed from the first and last columns of table 3a. We can also affirm the the Hill VPC 1 method is significantly superior to Hill VPC 2, despite their quite close results.

Also with a 95% confidence interval we can conclude that the Hill VPC 1 and 2 methods are superior to our *baseline* taking into account only VPC alignments as shown in the first column of table 3b. Hill VPC 2 is also superior to Giza++ refined (see last column of table 3b), but we are not able to conclude anything about the comparison of Hill VPC 1 and Giza++. These results hold for the Opus subtitle corpus on which our *gold standard* was based, widely different kind of corpora may show different behaviour.

5. Conclusion and Future Work

In this paper we proposed a discriminative lexical aligner for Portuguese and English that uses a traditional hill-climbing based discriminative core followed by language-dependent post-processing rules, for the handling of complex n:m alignments. We evaluated the approach on VPCs and the results show that it is possible to significantly improve alignment in all the relevant metrics in the test set. We also show that machine learning techniques can be very efficient in inducing the post-processing rules considered in this work.

Since dealing with MWEs is an important open problem in natural language processing, the approach proposed in this paper shows that it is possible to improve the alignment of at least some classes of MWEs using simple and effective shallow linguistic knowledge. To the best of our knowledge, existing efforts to deal with MWEs in the lexical alignment task (e.g. in [Venkatapathy and Joshi 2006]) have used only statistical information, while we combine statistical information generated by Giza++ with shallow linguistic knowledge using machine learning.

For future work, we intend to investigate the application of the same technique on other types of multiword expressions, such as light verbs or compound nouns. We also intend to add some statistical association measures to the feature vectors, in order to gauge if a combination of grammatical knowledge and statistical information can improve the results.

6. Acknowledgements

Acknowledgements go to CNPq, for having financed this research.

References

Baldwin, T. (2005). Deep lexical acquisition of verb-particle constructions. *Computer Speech and Language*, 19(4):398–414.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

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

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Caseli, H. M., Scalco, M. A. G., and Nunes, M. G. V. (2005). Annotation style guide for lexical alignment (nilc-tr-05-09). Technical Report 256. 24 p.

Deksne, D., Skadins, R., and Skadina, I. (2008). Dictionary of multiword expressions for translation into highly inflected languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 1401–1405, Marrakech, Morocco.

Evert, S. and Krenn, B. (2005). Using small random samples for the manual evaluation of statistical association measures. *Computer Speech and Language*, 19(4):450–466.

Fraser, A. and Marcu, D. (2006). Semi-supervised training for statistical word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 769–776, Sydney, Australia. Association for Computational Linguistics.

Germann, U. (2008). Yawat: Yet Another Word Alignment Tool. In *Proceedings of the ACL-08: HLT Demo Session*, pages 20–23, Columbus, Ohio. Association for Computational Linguistics.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Jackendoff, R. (2002). *English particle constructions, the lexicon and the autonomy of syntax*. Mouton de Gruyter, Berlin/New York, 1st edition.

Liu, Y., Liu, Q., and Lin, S. (2005). Log-linear models for word alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 459–466, Ann Arbor, Michigan. Association for Computational Linguistics.

Moore, R. C. (2005). A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Niehues, J. and Vogel, S. (2008). Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25, Columbus, Ohio. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ramisch, C., Schreiner, P., Idiart, M., and Villavicencio, A. (2008). An evaluation of methods for the extraction of multiword expressions. In *Proceedings of the LREC Workshop - Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 50–53, Marrakech, Morocco. Association for Computational Linguistics.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–15, London, UK. Springer-Verlag.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Somers, H. (1999). Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157.

Tiedemann, J. (2009). News from opus - a collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Venkatapathy, S. and Joshi, A. K. (2006). Using information about multi-word expressions for the word-alignment task. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 20–27, Sydney, Australia. Association for Computational Linguistics.

Zhang, Y., Vogel, S., and Waibel, A. (2004). Interpreting BLEU / NIST Scores : How Much Improvement Do We Need to Have a Better System ? In *Proceedings of Language Resources and Evaluation (LREC-2004)*, pages 2051–2054.

Zhao, Y. and Zhang, Y. (2008). Comparison of decision tree methods for finding active objects. *Advances in Space Research*, 41(12):1955 – 1959.