A Constraint Satisfaction Approach to Machine Translation

Sander Canisius[*,**] and **Antal van den Bosch**[*]
[*]Tilburg centre for Creative Computing, P.O. Box 90153, 5000 LE Tilburg, The Netherlands
[**]The Netherlands Cancer Institute, P.O. Box 90203, 1006 BE Amsterdam, The Netherlands
S.Canisius@nki.nl, Antal.vdnBosch@uvt.nl

Abstract

Constraint satisfaction inference is presented as a generic, theory-neutral inference engine for machine translation. The approach enables the integration of many different solutions to aspects of the output space, including classification-based translation models that take source-side context into account, as well as stochastic components such as target language models. The approach is contrasted with a word-based SMT system using the same decoding algorithm, but optimising a different objective function. The incorporation of source-side context models in our model filters out many irrelevant candidate translations, leading to superior translation scores.

1 Introduction

The vast complexity of the translation task has led designers of machine translation systems to delegate the task to multiple submodels. The crucial task in these systems is the integration of all available information in the best possible way. In this paper we advocate and present an eclectic, theory-neutral approach to machine translation that employs *constraint satisfaction* as the integrator method, and that regards the translation task as a *structured prediction problem*.

Structured prediction is a relatively new and emerging field in machine learning in which generic techniques are developed that explicitly model structural properties of the output space (Bakir et al., 2007). Statistical machine translation systems can be seen as a forebearer of this field; conditional random fields (Lafferty et al., 2001) and Searn (Daumé III, 2006) are more recent exponents of the approach. The typical solution to a structured prediction problem is to regard it as a combinatorial optimisation in an output space that spans all possible outputs for a given input. One or more learning components are responsible for learning (parts of) an objective function, and a search (or inference) component finds the output structure that maximises the objective function.

Within statistical machine translation systems, the probabilistic underpinning of all involved components acts as a kind of industrial standard. This has the positive effect that a substantial body of work could be built using the same universal language. At the same time, it makes the integration of non-stochastic components into statistical machine translation systems sometimes unwieldy. Yet, recent experiments on mixing local classifications of non-stochastic machine learners with statistical MT models (Carpuat and Wu, 2007; Stroppa et al., 2007), has shown its potential. Arguably, it makes sense to investigate the gathering of a mix of views on the objective function as input to the final search or inference process, and this mix should be eclectic—that is, theory-neutral—to allow extremely different but successful partial solutions to the objective function to participate.

As we argue in Section 2, the classic constraint satisfaction framework offers the right apparatus to offer such a theory-neutral basis. In Section 3 we describe our experimental setup; the outcomes of a comparative study with an unconstrained but otherwise equivalent word-level SMT system are discussed in Section 4. In Section 5 we formulate our conclusions.

2 Constraint satisfaction

In constraint satisfaction (Tsang, 1993) the goal is to find values for a set of variables that satisfy certain constraints. While a variable's domain dictates the values a single variable is allowed to take, the constraints of a constraint satisfaction problem specify which *simultaneous value combinations* over a number of variables are allowed. Here, we adopt a weighted constraint satisfaction approach. Candidate solutions to a weighted constraint satisfaction problem are scored according to the sum of weights of the constraints they satisfy, and the highest scoring solution is selected.

The constraint satisfaction approach presented here is formulated as an extension of statistical log-linear models (Berger et al., 1996; Papineni et al., 1998)[1]. A typical log-linear model for machine translation combines a number of feature functions, each of which measures the quality of a candidate translation according to some aspect. Two feature functions tend to be part of any SMT system; a translation model (TM), and a target language model (LM), measuring the faithfulness and the fluency of the translation, respectively. Both are probability distributions, obtained by maximum-likelihood estimation from training data. The best translation is determined by maximising a weighted sum of those feature functions:

$$\operatorname*{argmax}_{y} \lambda_{\text{TM}} \log P(x|y) + \lambda_{\text{LM}} \log P(y)$$

One of the problems with this traditional formulation is that the translation model ignores the context in which it is applied. This is the case for the source sentence context, as well as for the target sentence context. It is expected that the language model compensates for this. However, it is questionable whether this is a reasonable expectation. Since the language model does not take into account the source side at all, it can only resolve source-side ambiguities indirectly by looking at the translations of source words. This means that the language model is not only used for attaining good fluency, but also in part for attaining good faithfulness, the latter of which it might not be good enough for. Our extension uses constraint satisfaction to improve the translation model, by having it take into account both source and target sentence contexts.

To do this, we replace the language model by a *constraint model*. The score assigned to a candidate translation by this model corresponds to the sum of weights of satisfied constraints according to a constraint satisfaction problem. The score formula is adapted as follows:

$$\operatorname*{argmax}_{\mathbf{y}} \lambda_{\text{CM}} f_{\text{CM}}(\mathbf{y}) \quad \textbf{(1)}$$

$$+ \lambda_{\text{LM}} \log P(\mathbf{y}) \quad \textbf{(2)}$$

$$+ \lambda_{\text{NM}} \sum_{i} [y_i = \emptyset] \log P(y_i = \emptyset | x_i) \quad \textbf{(3)}$$

$$+ \lambda_{\text{LP}} |\mathbf{y}| \quad \textbf{(4)}$$

The constraint model feature function (1, CM) scores the satisfied soft constraints. Considering the difficulty of the translation task, we augment the objective function with three more feature functions. The language model (2, LM) is a standard back-off trigram language model, estimated using the SRILM toolkit (Stolcke, 2002). Two more feature functions are intended to compensate for the effect that n-gram language models tend to prefer shorter translations. The first, which we call the null model (3, NM), multiplies the translation probability of those source-language words that are translated to $\emptyset$, i.e. words for which no corresponding word is generated in the target-language sentence. It is estimated using relative frequencies, and is in fact similar to the translation model of SMT systems, with the exception that it only applies to source-language words left untranslated. The final feature function, the length penalty (4, LP), counts the number of target-language words. Given a positive weight λ_{LP}, it is in fact a length bonus rather than a penalty.

For the implementation of the constraint model, we define a weighted constraint satisfaction problem over a solution space of possible translations. Therefore, two questions need to be answered. First, how do we restrict the solution space? We aim at excluding most candidate solutions before the inference even starts. Defining this solution space is done by introducing variables and populating their domains, and by formulating certain hard constraints that every valid translation has to satisfy. Second, what soft constraints are added to the constraint satisfaction problem? We would like those constraints to improve the faithfulness of the translation by taking into account both source sentence context and target sentence context.

2.1 Solution space

Efficient approaches to machine translation have to make strong assumptions about the parts of the output space that are actually worth exploring. The approach presented here is sufficiently restricted

[1] The log-linear formulation of the objective function can be shown to be equivalent to a weighted constraint satisfaction problem, but we choose to follow this formulation, since it eases the comparison with SMT systems.

to allow for efficient decoding, while remaining expressive enough to attain good translation performance. To represent a solution space, we start by distinguishing two sub-problems that have to be solved as part of the translation task. First, source-language words have to be translated to the correct target-language word. Secondly, the translated words may need to be reordered—possibly new words have to be inserted as well—to make the translation a natural sentence according to the target language.

2.1.1 Representing word translations

For modelling the translation of words in the source sentence, we adopt the assumption that each word in the source sentence is translated to exactly one word in the target sentence. In our constraint satisfaction framework it is naturally represented by introducing one variable for each source word. During inference, the target-language words that are part of the domain of a variable will be considered as possible translations of the corresponding source word. If domains are constructed simply by listing all possible translations for the given source word as found in the training corpus, the solution space of our model would be rather similar to that of word-based SMT systems. In constraint satisfaction inference, however, we employ classifiers to predict the translations to consider. These predictions implicitly filter out all other possible outcomes, rendering the solution space potentially much smaller. We elaborate on how this is done later in the paper.

A few additional issues need to be dealt with. First of all, spurious words in the source sentence should not be translated to a target-language word. This is resolved by translating the word to a special $\emptyset$ symbol instead. By definition, this $\emptyset$ symbol will be part of the domain of all variables. As a result, any source-language word may be left untranslated in the target translation. A second issue is the fact that several source-language words might be translated by only one target word. In this case, all corresponding variables are assigned that same word. The fact that those matching words are actually a single token in the target sentence is dealt with in the target sentence realisation.

2.1.2 Representing target sentence realisation

Target sentence realisation involves three differences between the source language and the target language that have to be dealt with and represented: (1) word order differences; (2) zero-fertility words, and (3) multi-fertility words.

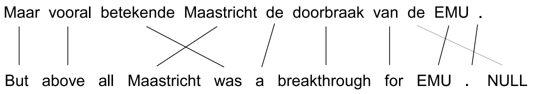
Maar vooral betekende Maastricht de doorbraak van de EMU .
But above all Maastricht was a breakthrough for EMU . NULL

Figure 1: Example of a Dutch-English aligned sentence pair.

Word order differences To cope with arbitrary word reorderings in the translation, the inference procedure needs to consider every permutation of translated source words. For a compact representation of the search space yielded as a result of this, consider a complete directed graph in which the words of the source sentence are represented by vertices, and one additional vertex v_0 corresponds to the start and end of the sentence. A directed arc from vertex v_i to vertex v_j means the translation of the word corresponding to v_j directly follows the translation of the word represented by v_i. In addition, a directed arc from v_0 to v_i, or from v_i to v_0 means that the translation of the word corresponding to v_i is the first or last word of the sentence respectively. The space of all candidate translations corresponds to all cycles that start and end at v_0. Such a cycle is not required to visit every vertex in the graph, i.e. it does not have to be a hamiltonian cycle. Cycles that do not visit a certain vertex v_i correspond to translation candidates in which the source word represented by v_i is not translated. In that case, the translation variable corresponding to this source word should have the value $\emptyset$, which can easily be enforced by a hard constraint.

Given this graph representation of the candidate translation space, it can be reformulated for the constraint satisfaction framework by introducing a set of $(n+1) \times (n+1)$ variables, where n is the length of the source-language sentence, that correspond to the adjacency matrix of the graph just introduced. The domains of all those variables comprise two values, signalling whether or not the corresponding arc is included in the candidate translation cycle. Appropriate constraints have to be added to the constraint satisfaction problem to ensure that every candidate considered is indeed a cycle of the graph. Informally, this is the case if for every $i \in \{1, 2, \ldots, n\}$, either the ith row and column do not contain any positive value at all, or they both contain exactly one positive value. Moreover, the 0th row and column *should* contain exactly one

	END	Maar	vooral	betekende	Maastricht	de	doorbraak	van	de	EMU	.
BEGIN		•									
Maar			•								
vooral					all						
betekende						•					
Maastricht				•							
de							•				
doorbraak								•			
van										•	
de											
EMU											•
.	•										

Figure 2: Visualisation of the connectivity matrix of the path corresponding to the correct translation of the Dutch sentence in Figure 1. The "•" in the row labelled with "Maar" and the column labelled with "vooral" denotes that the translation of the latter follows that of the former in the English target sentence. The word "all" is a zero-fertility word.

positive value. To illustrate all of the above, Figure 2 shows a matrix representing the correct translation order for the sample sentence in Figure 1.

Zero-fertility words A common approach to zero-fertility word insertion is to keep a list of frequent zero-fertility words and attempting to insert words from this list at arbitrary positions in the translation. This leads to a substantial expansion of the output space, which was already large to start with. As an alternative to common practice, we choose to attempt insertion of zero-fertility words only if there is evidence that doing so would make sense in the context of the current sentence. This evidence is to be provided by classifier predictions. More specifically, those predictions will be used to collect zero-fertility words that are candidates for insertion between the translations of two source words. In our representation at most one zero-fertility word can separate two fertile words.

Modelling this is possible by a straightforward extension of the adjacency matrix introduced before. In addition to the two values that signify whether or not two translated words are adjacent in the target sentence, a matrix element can also be assigned a word. Such an assignment encodes the case where two translated words are separated by the zero-fertility word stored in the matrix. In Figure 2, the matrix element that connects the translations of "vooral" and "Maastricht" has the value "all", which denotes that the translation of the latter follows that of the former, separated by the word "all". The constraints that ensure that only cycles of the order graph are considered as candidate solutions can be extended easily to this new setting. The zero-fertility word values can simply be treated as positive values.

Maar		But	above
vooral	But	above	all
betekende	Maastricht	was	a
Maastricht	all	Maastricht	was
de	was	a	breakthrough
doorbraak	a	breakthrough	for
van	breakthrough	for	EMU
de			
EMU	for	EMU	

Figure 3: The Dutch example sentence from Figure 1 and the English trigrams that are to be predicted for the words in the sentence. No training example is created for the word "de", because it is aligned with the ∅ token.

Multi-fertility words To account for many-to-one mappings, i.e. mappings of more than one source word to a single target word, we introduce one final value that can be assigned to order matrix entries, signalling overlap between two source words mapped to the same target word.

2.2 Constraints

We choose the constraints for machine translation to cover up to three consecutive target-language words. These constraints are created by predicting a trigram of target-language words for each word in the source sentence. Figure 3 illustrates this process for the sentence pair in Figure 1. The middle word of the predicted trigrams is the hypothesised translation of the source word in focus. The left and right parts are the words surrounding the translation in the target sentence. Note that no training example is created for the Dutch word "de" in Figure 1. Nevertheless, when translating a sentence, trigrams are predicted for all words in the source-language sentence—whether a word is aligned with ∅ is unknown for new sentences.

Given a predicted trigram, two types of constraints are extracted from it. First, a trigram constraint covering the translated word and the two words surrounding it in the target-language sentence. Secondly, two bigram constraints defined on the translated word and either one of the two surrounding words.

2.3 Solving the CSP

The solution space of the constraint satisfaction problem defined in this section has immense proportions. Even if a base classifier perfectly predicts the correct translations of all source words, which is already overly optimistic, the inference procedure still has to consider every possible permutation of those translated words as a candidate translation. Unfortunately, no further restrictions or assumptions can be made that would restrict the solution space sufficiently to allow for exhaustive solving. Approximate solving is the only option.

With this in mind, we choose the greedy decoding algorithm of Germann (2003) as the basis for the constraint solver. The algorithm starts with a complete candidate translation; for example, one where all source words are mapped to their most likely translations and added to the target sentence in original order. Subsequently, a hill-climbing search is started in which simple transformations of the current translation are attempted and the one leading to the highest score increase is actually applied. New transformations are tried until no further improvement can be attained. The following transformations are considered:

- *Change* the translation of a source-language word. If the target word currently aligned with it has a fertility greater than one, a new target word is inserted in the translation at the position maximising the translation score; otherwise, the current translation is changed, while its position is left unchanged. Among the translation candidates tried is also $\emptyset$, which results in the word being removed from the candidate translation.
- *Insert* a zero-fertility word. According to our model, zero-fertility words are only inserted in between two fertile words.
- *Erase* a zero-fertility word.
- *Join* two target-language words, i.e., removing one of the words from the translation and aligning with the remaining word all words previously aligned with the word that was removed.
- *Swap* two non-overlapping segments of the target sentence.

Although the algorithm has been proposed in the context of statistical machine translation, it can more generally be seen as optimising an arbitrary objective function defined over candidate translations. By replacing the noisy-channel equations optimised originally by a credit function based on constraint weights, the algorithm can be employed for solving our constraint satisfaction problem.

3 Experimental setup

3.1 Data

For our study we use four different corpora covering a diverse range of genres. From each of the four corpora, we prepare data sets for the translation pair Dutch to English:

EuroParl The EuroParl corpus (Koehn, 2005) is a multi-lingual parallel corpus extracted from the proceedings of the European Parliament. The Dutch-English parallel subcorpus consists of 1,313,111 sentence pairs.

JRC-Acquis The JRC-Acquis corpus (Steinberger et al., 2006) comprises a large collection of legislative texts extracted from the Acquis Communautaire. The Dutch-English parallel subcorpus provides 1,235,878 bilingual sentence pairs.

EMEA The EMEA data set is composed of texts made available by the European Medicines Agency. It is one of the corpora included in the OPUS parallel corpus (Tiedemann and Nygaard, 2004). The parallel texts for Dutch and English cover 751,602 sentence pairs.

OpenSubtitles The OpenSubtitles corpus, also part of OPUS, provides aligned movie subtitles in various different languages. For the language pair Dutch-English, it comprises 288,160 sentence pairs.

In four experiments, the translation system has been trained and tested on texts within the same corpus. For this evaluation, as well as for tuning the system, from each of the four corpora, two sets of 1,000 sentences each have been selected for testing and development purposes respectively; the remainder is used for training. This training data has subsequently been aligned at the word level using GIZA++ (Och and Ney, 2000).

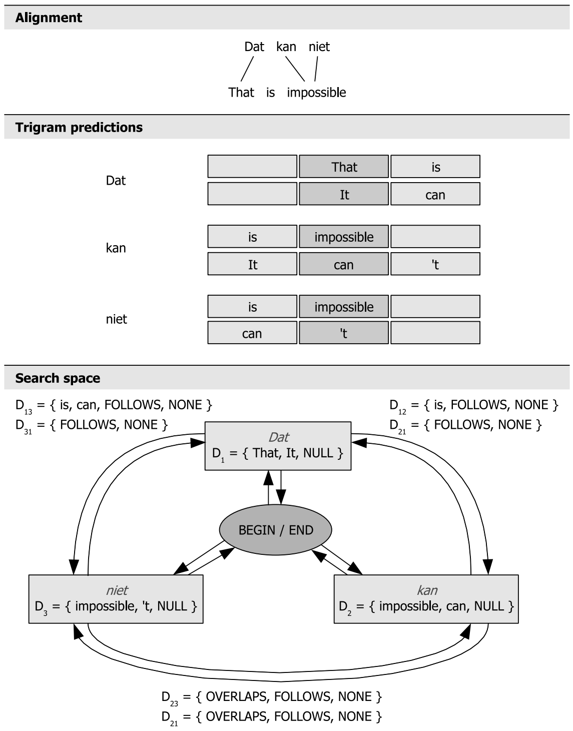

Figure 4: Visualisation of the search space resulting from a set of base classifier predictions. Top: the correct alignment of a Dutch-English example sentence pair. Middle: trigram predictions for the words in the Dutch sentence. Bottom: the complete graph connecting all Dutch words. Any valid translation is a directed cycle in this graph that starts and ends in the BEGIN/END node.

3.2 Constraint prediction

For predicting the soft constraints of our translation model, we need to map each word of the source sentence to a trigram of target words. The middle word of that trigram is the translation of the source word in focus; the left and right words are the two target words surrounding it.

Several recent studies (Carpuat and Wu, 2007; Chan et al., 2007; Giménez and Màrquez, 2007; Stroppa et al., 2007) have experimented with classification-based alternatives to traditional translation models that take into account contextual information of the word in the source sentence, similarly to the way word-sense disambiguation is performed. Our constraint predictor is similar to the classifiers used in these studies in the sense that contextual information is used to improve the suggested translations.

We follow (Stroppa et al., 2007) in using the k-nearest neighbor classifier as implemented in the TiMBL software package (Daelemans et al., 2007). The feature set used in our classifier is simpler, though. In specific, the features used correspond to a word window of length three centred on the focus word. As a consequence of the small number of features and the large number of classes, it will often be the case that the classifier finds several classes that have the same score for an input sentence. Classes that are assigned the same score by a base classifier are the perfect example of uncertainty that cannot be resolved locally, and thus should be delegated to the inference procedure. Therefore, for the experiments described in this paper, we disable tie-breaking in the base classifier, and extract domain values and constraints from all classes that have the maximum score.

The target-word trigrams predicted by the base classifier are used to add constraints to the inference, as well as to compose the domains of the variables. Constraints are derived from the predicted trigrams: the predicted trigram itself is turned into a constraint, but also the two bigrams covered by the predicted trigram.

The constraint satisfaction inference procedure is illustrated in Figure 4, where in the absence of tie-breaking in the classifier, two trigrams have been predicted for each source-language word. Since for all words, both trigrams suggest a unique translation, the domains of the three words, $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, each contain two candidate translations, as well as the symbol $\emptyset$, which is always included as a possible translation.

The variable domains for the order variables always contain at least the two values that signal that the two corresponding words do or do not follow one another in the translated sentence. In Figure 4, these variables correspond to the edges of the graph depicted at the bottom of the figure. The two symbols `FOLLOWS` and `NONE` are included in all domains. Furthermore, the model also allows for an overlap value or a zero-fertility word as value for order variables. As for the former, the overlap value is only added to the domain of the order variable y_{ij} if words i and j can be translated to the same target word, or more formally, if their domains overlap, $\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset$. As an illustration, $\mathcal{D}_2$ and $\mathcal{D}_3$ both contain the word "impossible", and therefore, $\mathcal{D}_{23}$ and $\mathcal{D}_{32}$ contain the symbol `OVERLAPS`.

Potential zero-fertility words are added to a domain only if base classifier predictions provide sufficient evidence for that. Specifically, the zero-fertility words included in the domain of the order variable y_{ij} are those words that appear both as the right part of the trigram predicted for word i, and as the left part of the trigram for word j. In the example, the words “That is” predicted for “Dat” overlap with the words “is impossible”, predicted for both “kan” and “niet”. For this reason, “is” is made a potential zero-fertility word if the translation of either “kan” or “niet” were to follow that of “Dat”. Similarly, “can” is a potential zero-fertility word between the translations of “Dat” and “niet”, since “It can” has been predicted for the former, and “can ’t” for the latter source-language word.

4 Results

To evaluate our constraint satisfaction approach to machine translation, we trained and tested the system using the four Dutch to English data sets described in Section 3.1. In addition, we implemented a word-based SMT system based on the ISI ReWrite decoder, which uses the greedy decoding algorithm of Germann (2003). Comparing with this system is especially interesting since the decoding algorithm is the same as the one used in our constraint satisfaction system. Therefore, the differences that are observed can be attributed to the modelling choices underlying the two systems. First, the constraint satisfaction system uses a richer objective function based on the constraint model that replaces the translation model. Second, constraint satisfaction inference searches a smaller solution space than the ReWrite system, which does not restrict its solution space in advance.

Table 1 lists the BLEU scores (Papineni et al., 2002) and exact Meteor scores (Banerjee and Lavie, 2005) for both systems on each of the four data sets. The two systems are closest in performance on the EuroParl data, though constraint satisfaction inference outperforms ReWrite in terms of both metrics. On EMEA, ReWrite outperforms constraint satisfaction inference; on JRC-Acquis and OpenSubtitles, constraint satisfaction inference outperforms ReWrite again.

The relative performance differences are rather diverse among the four data sets. This may be attributed to the underlying search algorithm, a greedy hill-climbing search, which is known to risk ending up in suboptimal local optima. Constraint satisfaction inference seems to deal with this circumstance better than ReWrite. On the one hand, the richer objective function used by constraint satisfaction inference, based on the predicted constraint model, may account for the better performance of constraint satisfaction inference. On the other hand, though, the smaller solution space searched by constraint satisfaction inference may also be expected to have fewer local optima.

The fact that on the EMEA corpus the ReWrite system performs better may be rooted in the fact that sentences in EMEA are largely formulaic and on average rather short: 9 tokens. Apparently, the hill-climbing algorithm only needs a few transformation operations to reach good translations. Constraint satisfaction inference’s objective function causes it to perform more transformation operations than it should.

5 Conclusions

Machine translation systems deal with huge output spaces that are costly to search. Their translation quality depends strongly on the quality of the inference imposed on the search in the output space. One strategy, presented in this paper, is to feed a theory-neutral inference mechanism, constraint satisfaction inference, with several different inputs of arbitrary types.

The decoding algorithm chosen for the experiments in this paper is an important ingredient for achieving the above objective. Since the algorithm is a local hill-climbing method, at any moment at which the objective function evaluates a hypothesis, there is a complete, rather than a partial translation as would be the case in A* or Viterbi search. As a result, the objective function can take into account arbitrary structural dependencies. The possibilities for such dependencies are virtually unlimited. In this paper, we experimented with only one type of constraint, which models trigrams of target-language words. We expect that large improvements can be achieved by introducing additional constraints. For example, constraints that model phrase-based translations, word reordering in the target sentence, or explicit syntactic structure of the target sentence.

A potential weakness caused by using a greedy search method is the risk of ending up with suboptimal solutions as a result of local optima in the search space. Although there is nothing that can really be done about this, one can make sure that the search space in which the decoder operates already has a certain minimum quality. Our constraint satisfaction inference approach uses a

	EuroParl		JRC-Acquis		EMEA		OpenSubtitles	
	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor
ReWrite	0.198	0.449	0.450	0.611	**0.395**	**0.650**	0.083	0.304
CSI	**0.211**	**0.469**	**0.513**	**0.650**	0.302	0.540	**0.200**	**0.444**

Table 1: BLEU and Meteor (exact) scores for constraint satisfaction inference and the ReWrite SMT system on the four Dutch to English translation tasks.

context-model classifier to define the exact solution space searched by the decoder. As the most important benefit of this, all candidate translations that are part of the solution space are predicted and filtered based on the context of the source-language word in the input sentence. The intended effect is that candidate translations that are irrelevant for the current sentence are not considered by the decoder, and thus local optima based on such translations are made impossible. Results from our comparative experiment show that this effect can indeed be attained.

Acknowledgements

This study was funded by the Netherlands Organisation for Scientific Research, as part of NWO IMIX and the Vici *Implicit Linguistics* project.

References

Bakir, G., T. Hofmann, B. Scholkopf, A. Smola, B. Taskar, and S. Vishwanathan. 2007. *Predicting Structured Data*. The MIT Press, Cambridge, MA.

Banerjee, S. and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Berger, A., S. Della Pietra, and V. Della Pietra. 1996. Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1).

Carpuat, M. and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72.

Chan, Yee Seng, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40.

Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2007. TiMBL: Tilburg Memory Based Learner, version 6.1, reference guide. Technical Report ILK 07-07, ILK Research Group, Tilburg University.

Daumé III, H. 2006. *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California.

Germann, U. 2003. Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the 2003 Human Language Technology Conference, NAACL-HLT 2003*, pages 1–8.

Giménez, J. and L. Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Lafferty, J., A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA.

Och, F.J. and H. Ney. 2000. Giza++: Training of statistical translation models. Technical report, RWTH Aachen, University of Technology.

Papineni, K, S. Roukos, and R Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. Intern. Conf. on Accoustics, Speech, and Signal Processing*, pages 189–192.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 2142–2147.

Stolcke, A. 2002. SRILM: An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904.

Stroppa, N., A. Van den Bosch, and A. Way. 2007. Exploiting source similarity for SMT using context-informed features. In *Proceedings of the 11th International Conference on Theoretical Issues in Machine Translation (TMI 2007)*, pages 231–240.

Tiedemann, J. and L. Nygaard. 2004. The OPUS corpus-parallel & free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 26–28.

Tsang, E. 1993. *Foundations of constraint satisfaction*. Academic Press, San Diego, CA.