Translating Negation: Induction, Search And Model Errors

Federico Fancellu and Bonnie Webber
School of Informatics
University of Edinburgh
11 Crichton Street, Edinburgh
f.fancellu[at]sms.ed.ac.uk , bonnie[at]inf.ed.ac.uk

Abstract

Statistical Machine Translation systems show considerably worse performance in translating negative sentences than positive ones (Fancellu and Webber, 2014; Wetzel and Bond, 2012). Various techniques have addressed the problem of translating negation, but their underlying assumptions have never been validated by a proper error analysis. A related paper (Fancellu and Webber, 2015) reports on a manual error analysis of the *kinds* of errors involved in translating negation. The present paper presents ongoing work to discover their *causes* by considering which, if any, are *induction*, *search* or *model* errors. We show that standard *oracle decoding* techniques provide little help due to the locality of negation scope and their reliance on a single reference. We are working to address these weaknesses using a chart analysis based on *oracle hypotheses*, guided by the negation elements contained in a source span and by how these elements are expected to be translated at each decoding step. Preliminary results show chart analysis is able to give a more in-depth analysis of the above errors and better explains the results of the manual analysis.

1 Introduction

In recent years there has been increasing interest in improving the quality of SMT systems over a wide range of linguistic phenomena, including coreference resolution (Hardmeier et al., 2014) and modality (Baker et al., 2012). Negation, however, is a problem that has still not been researched thoroughly (section 2).

Our previous study (Fancellu and Webber, 2015) takes a first step towards understanding why negation is a problem in SMT, through manual analysis of the kinds of errors involved in its translation. Our error analysis employs a small set of standard string-based operations, applying them to the semantic elements involved in the meaning of negation (section 3).

The current paper describes our current work on understanding the causes of these errors. Focussing on the distinction between *induction*, *search* and *model errors*, we point out the challenges in trying to use existing techniques to quantify these three types of errors in the context of translating negation.

Previous work on ascribing errors to induction, search, or model has taken an approach using oracle decoding, i.e. forcing the decoder to reconstruct the reference sentence as a proxy to analyse its potentiality. We show however that this technique does not suit well semantic phenomena with *local* scope (such as negation), given that a conclusion drawn on the reconstruction of an entire sentence might refer to spans not related to these. Moreover, as in previous work, we stress once again the limitation of using a single reference to compute the oracle (section 4.1)

To overcome these problems, we propose the use of an *oracle hypothesis*, instead of an *oracle sentence*, that relies uniquely on the negation elements contained in the source span and *how these are expected to be translated in the target hypothesis* at a given time during decoding (section 4.2).

Sections 5 and 6 report results of the analysis on a Chinese-to-English Hierarchical Phrase Based

Proceedings of SSST-9, Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 21–29,
Denver, Colorado, June 4, 2015.

Model (Chiang, 2007). We show that even if it possible to detect the presence of model errors through the use of an oracle sentence, computing an oracle hypotheses at each step during decoding offers a more robust, in-depth analysis around the problem of translating negation and helps explaining the errors observed during the manual analysis.

2 Previous Work

While recent years have seen work on automatically detecting negation in monolingual texts (Chowdhury and Mahbub, 2012; Read et al., 2012), SMT has mainly considered it a side problem. For this reason, no actual analysis on the type of errors involved in translating negation or their causes has been specifically carried out. The standard approach has been to formulate a hypothesis about what can go wrong when translating negation, modify the SMT system in a way aimed at reducing the number of times that happens, and then assume that any increase in BLEU score - the standard automatic evaluation metric used in SMT - confirms the initial hypothesis. Collins et al. (2005) and Li et al. (2009) considers negation, along with other linguistic phenomena, as a problem of *structural mismatch* between source and target; Wetzel and Bond (2012) considers it instead as a problem of *training data sparsity*; finally Baker et al. (2012) and Fancellu and Webber (2014) considers it as a *model problem*, where the system needs enhancement with respect to the semantics of negation.

Only a few efforts have tried to investigate errors occurring during decoding. Automatic evaluation metrics are in fact only informative about the quality of the output, but not about the decoding process that produces the output. As such, the most relevant related work are two studies on the main categories of errors during decoding (Auli et al., 2009; Wisniewski and Yvon, 2013). Both works use the reference sentence as a proxy to generate an oracle hypothesis but they differ in the technique they use and in the problem they are interesting analysing. Auli et al. (2009) targets induction errors — i.e. cases where a good translation is absent from the search space — by forcing the decoder to generate the reference sentence with varying translation options (for each source span) and distortion limits. If when increasing the number of target translations considered for each span, the number of references that is possible to fully generate also increases, an induction error has occurred. Results on a French-to-English PBSMT validates this hypothesis.

Wisniewski and Yvon (2013) considers instead oracle decoding as a proxy to distinguish search vs. model errors. If the oracle translation has a model score higher than the 1-best system output, a search error has occurred, since the system could not output the hypothesis with the highest probability; in contrast, a model error has occurred when the scoring function is unable to rank translations correctly. Here, the oracle translation is generated via ILP by maximising the unigram recall between oracle and reference translation, resembling the work of Germann et al. (2001) on optimal decoding in word-based models. In both Auli et al. (2009) and Wisniewski and Yvon (2013), almost all the errors during decoding are model errors.

A shortcoming of both methods is that neither can generate more than 35% of the references in the test set, by virtue of taking only one particular reference as the oracle, despite there usually being many ways that a source sentence can be translated.

3 Manual Error Analysis

This section briefly summarises the key points of the manual error analysis described in (Fancellu and Webber, 2015), since they also underpin the automated analysis described in section 4. The manual error analysis makes two assumptions:

- the semantic structure of negation can be annotated in a similar way across different languages, because the essentials of negation are language-independent.

- for analytic languages like English and Chinese, a set of string-based operations (*deletion*, *insertion* and *reordering*) can be used to assess translation errors in the semantics of negation.

Both assumptions involve first of all reducing a rather abstract semantic phenomenon into elements tangible at string-level. Following Blanco and Moldoval (2011), Morante and Blanco (2012) and Fancellu and Webber (2014), we decompose nega-

tion into its three main components, described below, and use them as the target of our analysis.

- **Cue**, i.e. the word or multi-words unit inherently expressing negation (e.g. 'He is not washing his clothes')
- **Event**, i.e. the lexical event the cue directly refers to (e.g. 'He is not washing his clothes)
- **Scope**, i.e. all the elements whose falsity would prove the statement to be true (e.g. 'He is not washing his clothes'); the event is taken to be part of the scope, since its falsity influences the truth value of negation. In the error analysis however, we exclude the event from the scope (since it is already considered *per se*) and further decompose the scope, to isolate the **semantic fillers** in its boundaries (*He*, *his clothes*), here taken to be Propbank-like semantic roles.

Given that we are combining standard, widely used error categories and language-independent semantic elements, we expect the annotation process and the error analysis to be robust and applicable to languages other than English and Chinese.

Results show an in-depth analysis of negation-related errors, where we are able to discern clearly which operations affect which elements and to what extent. We found the cue the element the least prone to translation errors with only four cases of it being deleted during translation. We also found reordering to be the most frequent error category especially for the fillers, given that the SMT system does not possess explicit knowledge of semantic frames and its boundaries.

By making use of the decoding trace, containing the rules used to build the 1-best hypothesis, we could also inspect the causes of deletion and insertion. We found that almost all deletion and insertion errors are caused by a wrong rule application that translates a Chinese source span containing negation into an English hypothesis that does not or vice versa. OOV items seem *not* to constitute a problem when translating negation. This is important especially in the case of the *cue*, whose absence means that the whole negation instance is lost. Given that all the cues in the test set have been seen during training, we also know the system has the ability to *potentially* reproduce negation on the target side.

4 Automatic Error Analysis

The manual error analysis can only get us as far as analysing the 1-best hypothesis and its building blocks. No explicit information on the causes of these errors can be recovered from the decoding trace only. To address this problem, we introduce two different techniques to analyse and distinguish different kinds of errors occurring at decoding time.

First however, we give a more formal definition of the three main categories of decoding-related errors as follows, where e and $p(e)$ are the optimal translation the decoder can produce, along with its probability while $\hat{e}$ and $p(\hat{e})$ stand for the 1-best output and its probability.

- **Search error**: $e \neq \hat{e}$ and $p(e) > p(\hat{e})$; the 1-best output is not the most probable output, given the model. Search errors are a consequence of the impossibility of exploring the entire search space, where more probable hypothesis may have been pruned.
- **Model error**: $e \neq \hat{e}$ and $p(e) < p(\hat{e})$; the model scores a semantically sub-optimal translation higher than the optimal one. This is because the scoring function lacks relevant features or the features present have not been properly weighted.
- **Induction error**: e cannot be generated because its components (phrases or rules) are absent from the search space.

4.1 Constrained Decoding

The first technique involves forcing the decoder to reproduce reference sentences if they contain negation. It reflects the assumption that if the system is able to reconstruct such oracles, it is *potentially* able to translate negation correctly.

We use the *constrained decoding* feature included in Moses (Koehn et al., 2007) to this purpose. In its basic implementation, constrained decoding assesses the degree of overlap between hypothesis and reference sentence; given a source span, the feature

function assigns a score to each of the target hypothesis as follows:

$$s_{constrDec} = \begin{cases} 1 & \text{if } \exists\, h \in H_p \wedge h \in R_p \\ -\infty & \text{if } \nexists\, h \in H_p \wedge h \in R_p \end{cases}$$

where h is a phrase in the hypothesis phrase set H_p and R_p is the set of reference phrases.

Constrained decoding can potentially reveal induction errors and distinguish between search and model errors. Following Auli et al. (2009), we try to increase the *translation option limit* parameter which determines how many target translations are considered for each source span; if larger values lead to the system being able to decode more references, induction errors are occurring. Using the same heuristics as Wisniewski and Yvon (2013), we can also distinguish between search error vs. model errors by checking whether the oracle has a total model score higher than the previous 1-best output or vice versa.

We also take into consideration the interaction between induction and search errors. A bigger search space would be needed in order to consider more target hypotheses per source span during decoding. Thus we experiment by combining different translation option limits and cube pruning pop limits, where the latter limits the number of hypotheses that can be inserted in each cell's stack, which in turn influences the size of the search space.

There is however a potential pitfall when applying these heuristics to the analysis of negation-related errors. Chances are in fact that negation does not scope over the entire reference sentence, as exemplified in (1), where only the first portion of the source and the last portion in the reference contain an instance of negation.

(1) *Src:* *jǐnánjūnqū* *mǒu*
Jinan military region some
bù *bànshì* *gōngkāi* *shǐ*
department business make public make
[*rèdiǎn*]$_{scope}$ *bù*$_{cue}$ *rè*$_{event}$
hot spots not hot
Ref: [Hotspots]$_{scope}$ not$_{cue}$ hot$_{event}$ due to transparent business procedures in Jinan military region

Given that negation can be (and usually is) a semantic phenomenon with a *local* scope, if the decoder fails to reproduce (1), one cannot simply conclude that negation-related elements cannot be reproduced. Moreover, because the oracle translation may involve elements outside the scope of negation, constrained decoding does not permit one to draw any conclusion about the kind of error that has occurred in the case of negation.

In order to overcome this problem, we try to isolate the elements of negation in both source and reference and run constrained decoding on those portions only. However, doing so demands we assume that negation is represented similarly in both source and the reference sentences. This is however not different from the general problem around *oracle decoding*, i.e. considering one reference sentence as the only ground truth. *Constrained decoding* is in fact an alignment problem, where we try to maximise the presence of reference segments in decoding, giving the source spans. If the reference spans are only paraphrases of the source spans, not direct translations, it is unlikely that the system will be able to reconstruct the oracle. Negation is not an exception, given the many ways that the same negation instance can be paraphrased. This is exemplified in (2) where the event is rendered in Chinese as an adjectival predicate (*lǐxiǎng* → 'ideal') while it is translated non-literally in the reference sentence as a nominal predicate ('*what it should be*').

(2) *Src:* [...] [*rénmen de jīngshén* *jiànkāng*
[...] people of psychology health
hěn]$_{scope}$ *bù*$_{cue}$ *lǐxiǎng*$_{event}$ [...]
very not ideal [...]
Ref: [...] [people's psychological health is]$_{scope}$ not$_{cue}$ [at all]$_{scope}$ what it should be$_{event}$ [...]

4.2 Chart Analysis

Constrained decoding demands the obviously false assumption that there is only one correct translation of a given source sentence. It also provides no alternative to assuming that conclusions formulated from those few references the system is able to reconstruct, also apply to the rest of the negated instances. Finally and most importantly, it is hard to explain the results obtained from the manual error analysis by simply reconstructing an oracle sentence and if it is really a case of model errors, there is no way

to know which model component (i.e. score) is the most responsible for a bad ranking of the hypothesis translations.

The approach we sketch out in this section tries to abstract from having a single reference and relies instead of *what is expected* to be translated at a given time during decoding. The end goal here is to compute *oracle hypotheses*, instead of *oracle sentences*.

We start by formulating four main expectations when translating instances of negation:

1. The **cue** has to be present
2. The **event** has to be correctly translated
3. The **cue** has to be attached to the correct **event**
4. The **fillers** have to be included in the right scope and connected to the right event in such way that they take the same (or an equivalent) semantic role to the one they had in the source.

Expectations (1) and (2) are related to the presence of a given element and allows us to analyse those instances of *deletions* observed in the manual error analysis; in (3) and (4), we investigate instead whether negation elements are grouped under the correct scope, therefore focusing on *reordering* errors.

If we know *at what time* during decoding we are translating a negation element, we can make use of these expectations; if a source sentence of length l contains a negation element in a span $S = s_n...s_m$ where $0 \leq n \leq m < l$ and given that cells in the decoding chart are indexed by the span they cover in the source, we expect that in cell [*i-j*], where $i \leq n \leq m \leq j$, the target hypotheses must contain a projection of this element and the two must be aligned.

Given these two assumptions, a comparison with constrained decoding is quite straight-forward. Meeting these expectations is the same as computing an oracle, but instead of doing it at sentence level, we do that at a hypothesis level (hence the name *oracle hypothesis*), that is, for each covered span in the source (here taken to be a cell in the chart).

The scores for each hypothesis in the cell provide detailed information about the presence of model errors; since we expect hypotheses that satisfy the four expectations above to be scored (and ranked) higher than those which do not, we can not only calculate the number of times this is not the case, but we can only see how low in the rankings a good translation is and which features cause this failure. By varying the *translation options limit* and the *cube pruning pop limit* parameter, we can also investigate whether these expectations are not met because of search and induction errors. Even if the search space is so vast that it is practically impossible to explore it all, we assume that with a large upper bound of hypotheses per stack, we are able to capture all relevant errors, and if any are not captured, they can be attributed to the "long tail" of rare occurrences.

The main two challenges at this point are to know (a) which elements in the source are negation elements and (b) whether they are translated correctly in the target hypothesis. In the case of (a) we use the manual annotation presented in (Fancellu and Webber, 2015). Future work will try to automate the process.

Challenge (b) requires a way to compute those expectations on the target (English) side. In order to detect the presence of a cue, we build a list of English negation cues from the training data using the exact same heuristics and training data as Chowdhury and Mahbub (2012) and check whether a given hypothesis contains a cue from this list. In order to deal with those cases of *lexical negation* where cues in the source are rendered as part of the meaning of a word in the target (e.g. zh: ***bù***tóng → en: '*different*'), we extract a mapping between Chinese cues and these words covertly expressing negation from the manually aligned GALE Chinese-English Word Alignment and Tagging Training data (Li et al., 2012).

In order to recognise the presence of a correct event, it is possible to check whether the hypothesis contains a good translation of the source using bilingual dictionaries (e.g. CCEDIT[1]) and enriching the results through synonyms (e.g. WordNet) and paraphrases databases (e.g. PPDB (Ganitkevitch et al., 2013)).

To ensure that the cue refers to the right event, we use the Stanford dependency parse (Manning, 2008) and apply it to each of the target (English) hypothesis in the cell's stack to check whether a subordinate-head relation is established between the two. Given

[1] http://www.mdbg.net/chindict/chindict.php?page=cedict

that the Stanford parser does not build a *neg* relationship from each negation cue to its head event, we just check more in general whether the cue is in a subordinate relationship with the event.

Finally, we use the dependency parse to verify that the fillers are correctly connected to negated event. This is a problem that needs more consideration and is therefore left for future work. The correct rendering of the fillers in the negation scope is in fact related to the more general open-problem of preserving predicate-argument structure during translation.

We are also exploring a second approach where we detect these elements on the English side by generating as many paraphrases as possible from the reference sentences using the same approach of (Zhao et al., 2009) and the PPDB database. We then extract cues, events and fillers from these paraphrases automatically and check whether they are present in the chart hypotheses and they correctly relate to each other.

5 System

We carried out the error analysis on the output of the Chinese-to-English hierarchical phrase based system submitted by the University of Edinburgh for the NIST12 MT evaluation campaign. The system was trained on ∼2.1 millions length-filtered segments in the news domain, with 44678806 tokens on the source and 50452704 on the target, with MGIZA++ (Gao and Vogel, 2008) used for alignment. The Chinese side of the training and the test set were segmented using the LDCWordSegmenter. The system was tuned using MERT (Och, 2003) on the NIST06 set.

The automatic error analysis was carried out on a sub-set of 54 segments the NIST MT08 test set[2], each containing at least an instance of negation on the source side. Although small, this set was considered to be representative given that it clearly shows a pattern in the errors involved in translation negation.

[2]This sub-set containing only negative sentences was extracted during the manual evaluation. Out of 1357 segments in the NIST MT08 set, we randomly picked 250 segments and annotate all instances of negation whether present

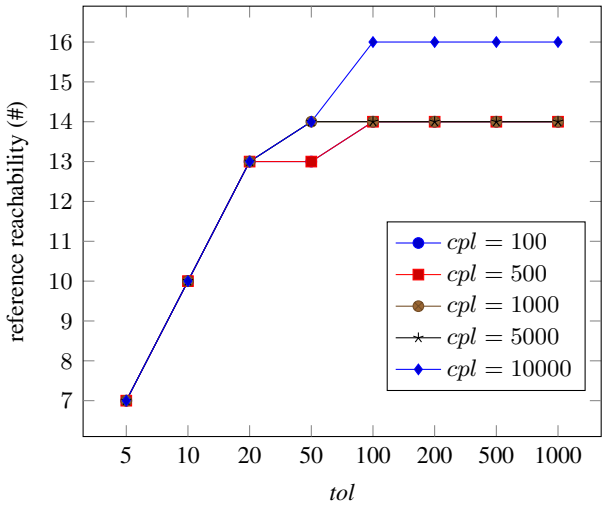

Figure 1: Number of reachable oracle negation instances plotted against the *translation option limit* (*tol*) for each of the five *cube pruning pop limit* (*cpl*).

6 Results

In this section we present the results related to the two methods introduced in sect. 4.

As shown in Figure 1, given the default settings of our decoder (*tol*: 20; *cpl*: 1000), we were able to generate only 13 out of 54 references in the test set (24%). Increasing the *translation options limit* to 100 leads only to a slight improvement and an upper bound of 16 reachable references (29%). We also did not see any noteworthy interaction between *translation option limit* and *cube pruning pop limit* (where *cpl* values of 500 and 1000 track the graph for a *cpl* of 5000); if there is no need for a large number of hypotheses to be considered during decoding to reconstruct the reference, there is also no need for a bigger search space.

Finally, comparison of the total model score of the oracle hypothesis vs. the 1-best output shows that in all cases the score of latter is higher than the former. We can conclude that for the references the system was able to reconstruct, model errors are a major cause of failure whilst induction and search errors are not. However, the number of references the system was able to fully reconstruct is very low, which makes it hard to draw final conclusions from constrained decoding alone, including any connection between these results and our manual error analysis. We present here preliminary results for the chart

analysis approach. We focus on detecting an *oracle hypothesis* that contain a right translation of the *cue* (therefore satisfying only expectation 1 in sect. 5.2).

Our first goal is to identify those cases where the cue is absent in the final cell, since deletion was the only type of error that involved the cue. However, in general, we want to have a measure of how strong our model is when translating the negation cue. A good model should in fact always be able to correctly translate a cue whether present in the source span.

We found that there are a total of 14948 cells for the whole test set where a translation of the cue is expected (i.e. the source contains a cue in the span the cell covers), for an average of ∼277 cells per sentence. We found that in 8311 of those cells (∼57%), a projection of the cue is absent, four of which are final, meaning that the cue is absent from four of the hypothesis translations output by the system. However, a per sentence distribution of the cells where the cue is expected but absent (Figure 2) shows that there is at least one cell in a chart containing the correct cue. Conversely, in no chart is the cue is completely absent. This means that in all cases the cue was reproduced at same point but in some, it failed to propagate to the final cell. This shows that chart analysis is useful to explain those cases of cue-related errors found in the manual analysis. We can conclude that the system is always *potentially* able to translate the cue. Given that there is no shortage of rules to translate the cue with default parameters, we can also conclude that, for the negation element here considered, no induction error has occurred. This conclusion is more solid than the one drawn from the constrained decoding approach, since it is based on the analysis of the decoding process for the entire test set.

We also found that in each cell an hypothesis containing the right translation of the *cue* is, on average, ranked highly (2.79, where 0 represents the 1-best hypothesis). Out of the 1100 cases where the 1-best hypothesis and the cue-translation *oracle hypothesis* are not the same, the times the scores of the former are higher than the latter are: 275 for LM score (25%), 730 for the indirect translation probability (**66**%), 718 for the indirect lexical probability (**65.2**%), 525 for the direct translation probability (47.7%) and 435 for the direct lexical probability

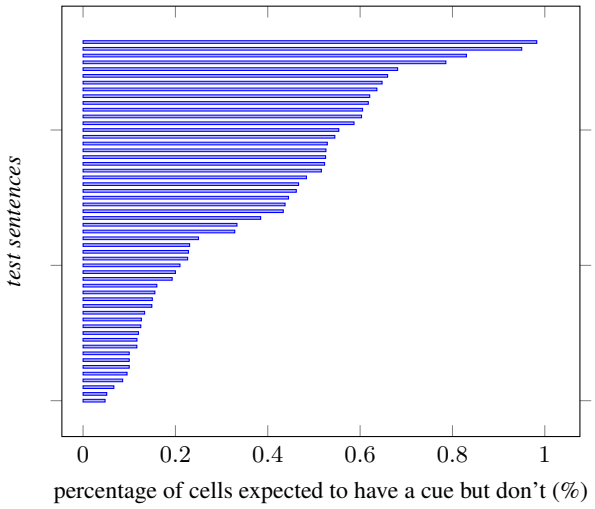

Figure 2: Distribution of cells per sentence *not* containing the expected *cue*.

(39.5%). Chart analysis can show us which features are the most responsible for *model errors*. We found out that the translation model holds the main responsibility for incorrectly ranking hypotheses containing the correct cue projection. Again, this useful form of analysis could not have been carried out using *constrained decoding* alone.

Finally, we are left to consider the impact of *search errors* in translating the negation cue. We first check whether these are involved in the four cases in which the cue is absent from the system's output translation, by testing with larger *cube pruning pop limit* and *translation options limit* values. Results shows that even by considering large values (of 10000 and 1000 respectively), no cue translations was found in the final cell of the chart for those sentences where deletion occurs. Enlarging the *search space* does not lead to any more cue translations making it to the final cell of the chart, highlighting the fact that translation of cue does not involve *search errors*.

7 Conclusion

In the present paper, we presented ongoing work on analysing the *causes* of the errors involved in translating negation, targeting three main categories: **induction**, **search** and **model** errors.

Following previous work, we applied an *oracle decoding*-based technique to detect those errors

by forcing the decoder the generate the reference sentence. Conclusions drawn from the references the decoder could reconstruct show that translating negation primarily involve model errors. However, the technique has two important limitations: (a) drawing conclusion from the reachability of an entire reference sentence is not informative when analysing semantic phenomena that usually have a *local* scope, such as negation; (b) the oracle is taken to be one reference sentence, while there are usually many ways to translate a sentence correctly and therefore (c) the results obtained applies to only part of the test set and cannot be taken to represent the entire data; (d) being able to generate an oracle does not give any in-depth insight on the each decoding step which is detrimental if we have to explain the results from the manual analysis.

Given these shortcomings, we sketch out an analysis that is able to compute partial *oracle hypotheses*, given the negation elements contained in a source span and four main *expectations* related to how negation elements should be translated at a given time during decoding. Preliminary results on *cue* translation show that the system can *potentially* translate all the cues in all the test sentences. No induction or search errors were found meaning that *model errors* are the only category of errors occurred in translating the negation cue. Moreover, a comparison between 1-best and *oracle* hypotheses show that the translation model scores are the main responsible for bad ranking. In general, it was shown that our method is able to give a more in-depth analysis of the process of translating negation at decoding time.

8 Future Work

In the present work, we have only presented the general idea around considering *oracle hypotheses* instead of *oracle sentences*, along with some preliminary results. Further work is however necessary to complete the analysis of the other two elements of negation – **event** and **fillers**.

It is worth remembering several factors can impact the kind of errors found in translation. Hierarchical phrase-based models are in fact non-purely syntax driven methods that are able to deal with high levels of reordering. That however also means that (a) there is no concept of constituent boundaries and (b) when reordering is performed incorrectly there is a high degree of element scrambling. We therefore accept that system-related proprieties might influence the presence of one error class over another and it will therefore be useful to conduct the same analysis on different models. In the same way, different languages will also display different problems and it is therefore necessary to consider the choice of language pair as another variable that can influence the result of such analysis.

9 Acknowledgements

This work was supported by the Accept, MosesCore and GRAM+ grants. The authors would like to thank Adam Lopez for his comments and suggestions and the two anonymous reviewers for their feedback.

References

Auli, M., Lopez, A., Hoang, H., and Koehn, P. (2009). A systematic analysis of translation model search spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 224–232.

Baker, K., Bloodgood, M., Dorr, B. J., Callison-Burch, C., Filardo, N. W., Piatko, C., Levin, L., and Miller, S. (2012). Modality and negation in SIMT use of modality and negation in semantically-informed syntactic MT. *Computational Linguistics*, 38(2):411–438.

Blanco, E. and Moldoval, D. (2011). Some Issues on Detecting Negation from Text. In *Proceedings of the 24th Florida Artificial Intelligence Research Society Conference (FLAIRS-24)*, pages 228–233, Palm Beach, FL, USA.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational linguistics*, 33(2):201–228.

Chowdhury, M. and Mahbub, F. (2012). FBK: Exploiting phrasal and contextual clues for negation scope detection. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 340–346.

Collins, M., Koehn, P., and Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540.

Fancellu, F. and Webber, B. (2014). Applying the semantics of negation to SMT through n-best list re-ranking. *EACL 2014*, page 598.

Fancellu, F. and Webber, B. (2015). Translating negation: A manual error analysis. In *Workshop On Extra-Propositional Aspects of Meaning (ExProM) in Computational Linguistics - NAACL '15*.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 228–235.

Hardmeier, C., Tiedemann, J., and Nivre, J. (2014). Translating pronouns with latent anaphora resolution. In *NIPS 2014 Workshop on Modern Machine Learning and Natural Language Processing*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180.

Li, J.-J., Kim, J., Kim, D.-I., and Lee, J.-H. (2009). Chinese syntactic reordering for adequate generation of Korean verbal phrases in Chinese-to-Korean SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 190–196.

Li, X., Grimes, S., and Strassel, S. (2012). Gale chinese-english word alignment and tagging training part 3. Technical report, Philadelphia: Linguistic Data Consortium.

Manning, C. (2008). Generating typed dependency parses from phrase structure parses.

Morante, R. and Blanco, E. (2012). *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 265–274.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167.

Read, J., Velldal, E., Øvrelid, L., and Oepen, S. (2012). Uio 1: constituent-based discriminative ranking for negation resolution. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 310–318.

Wetzel, D. and Bond, F. (2012). Enriching parallel corpora for statistical machine translation with semantic negation rephrasing. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 20–29.

Wisniewski, G. and Yvon, F. (2013). Oracle decoding as a new way to analyze phrase-based machine translation. *Machine translation*, 27(2):115–138.

Zhao, S., Lan, X., Liu, T., and Li, S. (2009). Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 834–842.