ProphetMT: Controlled Language Authoring Aid System Description

Xiaofeng Wu, Liangyou Li, Jinhua Du, Andy Way
ADAPT Centre, School of Computing,
Dublin City University, Ireland
xiaofengwu,liangyouli,jdu,away@computing.dcu.ie

Abstract

This paper presents ProphetMT, a monolingual Controlled Language (CL) authoring tool which allows users to easily compose an in-domain sentence with the help of tree-based SMT-driven auto-suggestions. The interface also visualizes target-language sentences as they are built by the SMT system. When the user is finished composing, the final translation(s) are generated by a tree-based SMT system using the text and structural information provided by the user. With this domain-specific controlled language, ProphetMT will produce highly reliable translations. The contributions of this work are: 1) we develop a user-friendly auto-completion-based editor which guarantees that the vocabulary and grammar chosen by a user are compatible with a tree-based SMT model; 2) by applying a shift-reduce-like parsing feature, this editor allows users to write from left-to-right and generates the parsing results on the fly. Accordingly, with this in-domain composing restriction as well as the gold-standard parsing result, a highly reliable translation can be generated.

Keywords: Controlled Language, Authoring Tool, Statistical Machine Translation

1. Introduction

Although current machine translation (MT) methods have improved rapidly in the past decade, SMT is still not reliable enough to be considered human-quality without significant post-editing (O'Brien, 2005). The primary reason is that natural languages are full of ambiguities.

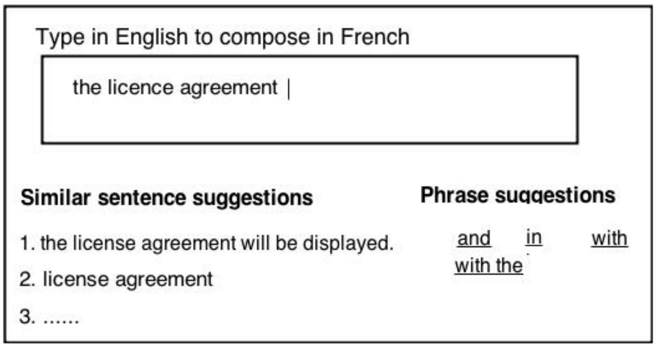
A Controlled Language (CL) is widely used in professional authoring where the aim is to write for a certain standard and style demanded by a particular profession, such as law, medicine, patent, technique etc (Gough and Way, 2004; Gough and Way, 2003). For multilingual documents, CL has been shown to improve the quality of the translation output, whether the translation is done by humans or machines (Nyberg et al., 2003).

The advantages of applying CL are self-evident: clear and consistent composition guidelines as well as less ambiguity in translation. However, the problems are also obvious: design of the rules usually requires human linguists, and rules may be difficult for end-users to grasp. In addition, the sentences that can be generated are often limited in length and complexity (O'Brien, 2003).

This paper presents ProphetMT,[1] a tree-based SMT-driven CL authoring tool. ProphetMT employs the source-side rules in a translation model and provides them as auto-suggestions to users. Accordingly, one might say that users are writing in a 'Controlled Language' that is 'understood' by the computer.

2. Related Work

All existing computer-aided authoring tools within a translation context employ a kind of interactive paradigm with a CL. Mitamura (1999) allow users to compose from scratch, and discuss the issues in designing a CL for rule based machine translation. Power et al. (2003) describes a CL authoring tool for multilingual generation. Marti et al. (2010) present a rule-based rewriting tool which performs syntactic analysis. Mirkin et al. (2013) introduce a confidence-driven rewriting tool which is inspired by Callison-Burch et al. (2006) and Du et al. (2010) that paraphrases the out-of-vocabulary words (OOV) or the "hard-to-translate-part" of the source side in order to improve SMT performance.



Type in English to compose in French

the licence agreement |

Similar sentence suggestions

1. the license agreement will be displayed.
2. license agreement
3. ......

Phrase suggestions

and in with with the

Figure 1: SMT-driven Authoring Tool by (Venkatapathy and Mirkin, 2012)

To our knowledge, Venkatapathy and Mirkin (2012) is the first interface that could be called an SMT-driven CL authoring tool: the main interface screen shot is shown in Figure 1. Their tool provides users with the word, phrase, even sentence level auto-suggestions which are obtained from an existing translation model. Nevertheless, it lacks syntactically-informed suggestion and constraints.

Sentences in all languages contain recursive structure. Synchronous context-free grammars (SCFG) (Chiang, 2005) and stochastic inversion transduction grammars (ITG) (Wu, 1997) have been widely used in SMT and achieve impressive performance. However, MT systems which make use of SCFG tend to generate an enormous phrase table containing many erroneous rules. This huge search space not only leads to an unreliable output, but also restricts the input sentence length that the system can handle. Other tree-based SMT models like Liu et al. (2006) and shen2008new depend heavily on the accuracy of the parsing algorithm

[1] ProphetMT has been granted Enterprise Ireland Feasibility Study Funding 2016.

which introduces noise upstream to the MT system.

Our method, ProphetMT, allows monolingual users to easily and naturally write correct in-domain sentences while also providing the structural metadata needed to make the parsing of the sentence unambiguous. The set of structural templates is provided by the tree-based MT system itself, meaning that highly reliable MT results can be generated directly from the user's composition.

Syntactic annotation is a tedious task which has traditionally required specialised training. In order to maintain a natural and easy writing style, ProphetMT makes use of auto-suggestion both for syntactic templates and for terms. A shift-reduce like (Aho, 2003) authoring interface, which allows users to easily parse the "already composed part" of the sentence, is also applied to maintain the structural correctness and unambiguous parsing while the source sentence is being composed.

3. ProphetMT: Syntactical SMT-Driven Authoring

3.1. An Overview of ProphetMT

ProphetMT is a client-server application. There are three main components involved:

1. A website client provides a structural writing user interface.
2. A web-service provides source-language rule/term auto-completion.
3. A web-service provides hierarchical phrase-based machine translation.

The main interface is shown in Figure 2. The 4 areas are:

1. the input area (upper)
2. the source tree structural area (middle left)
3. the target tree structural area (middle right)
4. the composed sentence and the translation (bottom)

The behavior of the ProphetMT can be defined by Algorithm 1, explained below are some terminology:

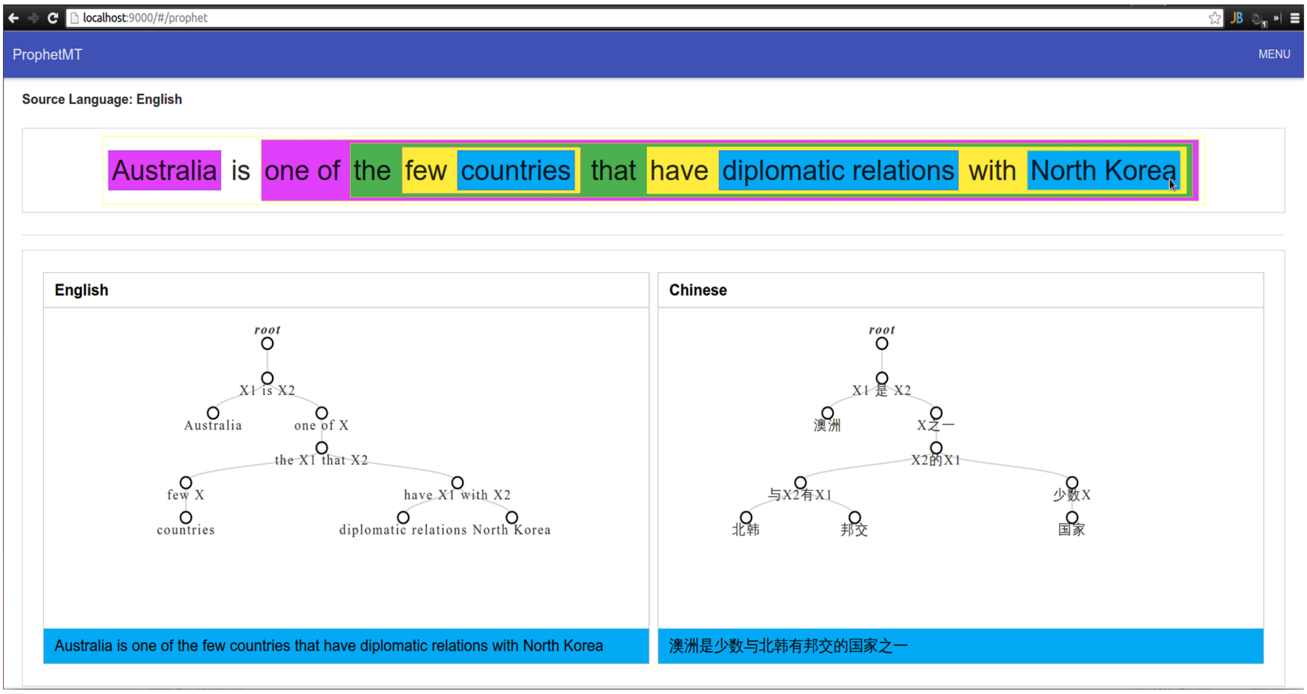
- NodeBox: the recursive (nestable) editing unit
- Non-Terminal Rule (NTR): rules like "*X is X*", "*one of X*", "*has X with X*" which have variables
- Non-Terminal(NT): the "*X*" in the NTR
- Terminal Rule(TR): rules without NT

While the user is inputting text, both TR auto completion and NTR auto completion are provided. Autocompletion candidates are automatically selected from the normal tree-based MT model according to the guidance introduced in Section 4.. When the user finishes composing the sentence, the result is sent to a tree-based MT engine, and the target translation(s) are generated according to the source- side rules decided by the user.

With the following example, we further explain Algorithm 1 in detail.

3.2. An Example

Suppose the user wants to input the sentence "*Australia is one of the few countries that have diplomatic relationships with North Korea*", which is shown in Figure 3 together with the NodeBox numbers.



Source Language: English

Australia is one of the few countries that have diplomatic relations with North Korea

English

Australia is one of the few countries that have diplomatic relations with North Korea

Chinese

澳洲是少数与北韩有邦交的国家之一

Figure 2: ProphetMT Main Interface Screen Shot

```
Initialize: ProphetMT opens an empty NodeBox;
while User is typing in an NodeBox do
    Provide TR auto suggestions;
    if There is a left adjacent NodeBox then
        Provide all NTR suggestions;
    else
        Provide the NTRs which DO NOT have NT at the
          beginning position;
    end
    if User selects "translate" then
        Finish the source and target parse trees;
        Translate and output the results;
        Go to stop;
    end
    if User Chooses an NTR then
        Generate the according NodeBoxs;
        if The current selected NTR has an NT at the
          beginning position then
            The corresponding NodeBox will
              automatically merge with the left adjacent
              NodeBox ;
        end
        Focus goes to the first NodeBox which is empty;
        Continue;
    end
    if User starts a new NodeBox then
        Stop the current NodeBox editing;
        Focus goes to the new NodeBox;
        Continue;
    end
end
Stop:
```

Algorithm 1: ProphetMT Main Workflow

The following steps will be performed:

1. ProphetMT starts a new NodeBox1 andthe user types in "*Australia*"; NodeBox1 is finished.
2. User starts a new NodeBox0 to the right of the NodeBox1 and types in "*is*"; The user chooses an NTR "*X is X*" , then two new NodeBoxes will be generated within NodeBox0 by "*NodeBox is NodeBox*";The left NodeBox within NodeBox0 will automatically **merge** with the NodeBox1 which is left adjacent to NodeBox0; NodeBox0 is finished.
3. The user selects the second NodeBox within NodeBox0, which is NodeBox2, and types in "*one of*" and selects the NTR "*one of X*". NodeBox2 is finished.
4. In the generated NodeBox3, th user types in "*the*" and chooses "*the X that X*". The NodeBox3 is finished and two new NodeBoxes are generated as NodeBox4 and NodeBox6
5. In NodeBox4, the user types in "*few*" and chooses "*few X*". NodeBox4 is finished, NodeBox5 is generated.
6. In NodeBox5 the user types in "*countries*". NodeBox5 is finished
7. In NodeBox6, the user types in "*have*" and chooses NTR "*have X with X*". NodeBox7 is finished, two new NodeBoxes, NodeBox7 and NodeBox8 are generated.
8. In NodeBox7, the user types in "*diplomatic relations*". NodeBox7 is finished.
9. In NodeBox8, the user types in "*North Korea*". NodeBox8 is finished.
10. The user finishes editing the sentence and then the translation for the specific languages as well as the parsing trees are generated.

3.3. Merging

The merging process which happens in step 2 is shown in Figure 4. This merging process allows the user to compose the sentence from left-to-right while keeping the partially parsed structure intact.

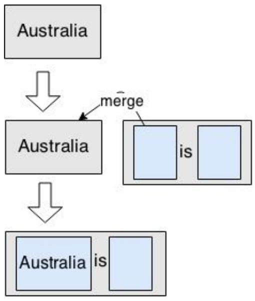


Figure 4: The Merging Process

3.4. NodeBox Starting Points Selection

Figure 5 further illustrates how the user starts a new NodeBox and how ProphetMT maintains the syntactic structure by adopting a shift-reduce-like strategy. Suppose the user has written "Australia ... and China ...". Figure 5a shows the current state in the input area and the arrows "A" and "B" are the possible inserting point options. Figure 5b shows the corresponding partially parsed tree shown in the source parsing area which also indicates the two insertion positions. Figure 5c shows the parsing area when the user wants to further describe China and chooses the rule "*X which is X*" in position "A". Because there is a left-adjacent NodeBox and there is a NodeBox at the start position of the selected rule, so a merging process takes place. Figure 5d

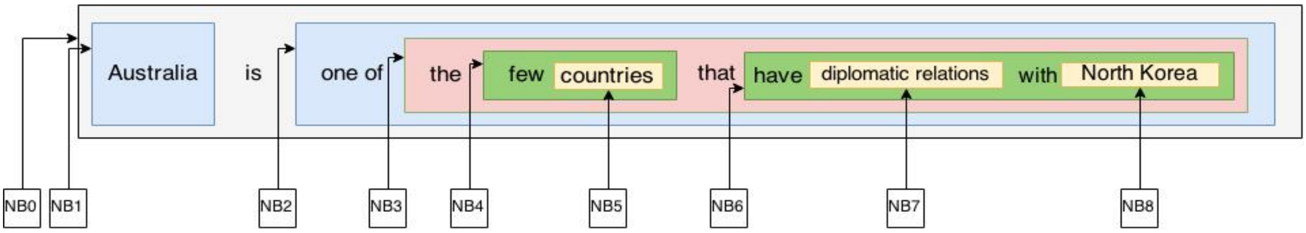
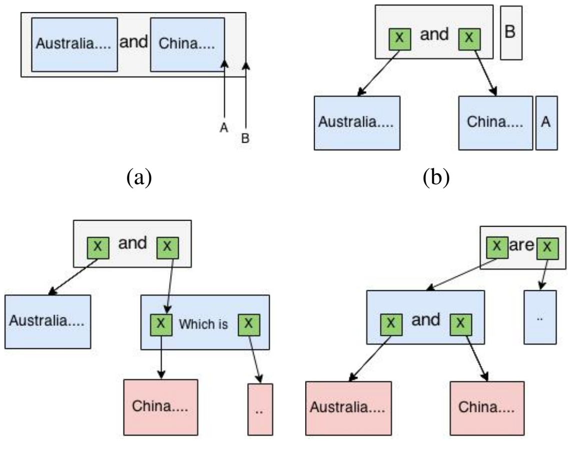


Figure 3: Example of Using ProphetMT Together With The NodeBox Numbers

shows the parsing area when the user wants to keep "Australia .. and China .." as a unit and chooses the rule "*X are X*" at position "B". As shown, a similar merging process happens.
We can see that this shift-reduce strategy allows composition to proceed from left-to-right while at the same time maintaining a correct parse of the existing text.



(a) (b)

(c) (d)

Figure 5: NodeBox Starting Points Selection

4. Auto-suggestions

In this section we introduce the auto-suggestion web-service employed in ProphetMT, including the *phrase level*, *rule level*, and *paraphrase* suggestion engines.

4.1. Terminal Rule (Phrase) Auto-suggestions

Following the work of Venkatapathy and Mirkin (2012), the phrase level auto-suggestions are also guided by three factors:

- **Fluency**: What the user has input will influence the incoming auto-suggestion. We use the SRILM (Stolcke and others, 2002) toolkit to rescore the phrase auto-completion.
- **Translatability**: The phrase pairs in the phrase table (i.e. the SMT model) are be sorted according to the four translation possibility features.
- **Semantic Distance**: The semantic distance of the suggested phrases must be close to the already composed part.

The final rank of the proposed phrases is based on the minimization of the Semantic Distance and maximization of the Fluency and Translatability.

4.2. Non-Terminal Rule (NTR) Auto-suggestions

In order to extract NTRs which are meaningful to humans, we parse the source side of the training corpus with the Berkeley Parser[2]. Then we wrap the parsed result with xml for Moses[3] (Koehn et al., 2007) hierarchical phrase-based model to extract rules. The ranking of NTR suggestions follows the same methodology that was employed for phrase suggestions.

4.3. Filtering

To further reduce the size, the NTRs containing content words like nouns, pronouns and numbers can be removed. This filtering is based on the observation that the structure of a sentence is primarily dictated by function words as well as verbs. The phrase-level auto-suggestions are responsible for providing the content words that fill the leaf nodes in the hierarchical templates. Because most NTRs will be discarded, and because the source side is already parsed when fed to the decoder, the normal restrictions of tree-based models, such as the maximum span (which is $<= 20$) and the NT numbers (which is usually $<= 2$), can be removed.

4.4. Paraphrase Auto-suggestions

If the user inputs an OOV, a paraphrase engine will be queried to try to suggest terms within the current SMT model. Paraphrases are obtained from PPDB.[4] If the OOV is not found in PPDB, than the user will be forced to choose another word.

5. Tree-based SMT

When the user finishes composing the sentence and commands ProphetMT to translate, a tree-based SMT engine will take the sentence and the user parsing results as input to generate the final target language selected by the user.

6. Preliminary Evaluation

Currently we have the UI implementation and the backend server. In this section we provide a preliminary evaluation. The following example was downloaded from Moses.[5] We use the model in the tree-to-tree folder and replace the non-terminal tag with "X" in order to be compatible with Moses hierarchical phrase-based (HPB) model. The whole model is shown below.

```
overhead [X] ||| toudingshangde [X]
overhead [X][X] [X] ||| toudingshangde [X][X] [X]
masks [X] ||| mianzhao [X]
oxygen [X] ||| yangqi [X]
[X][X]1 in the [X][X]2 [X] ||| [X][X]2 de [X][X]1 [X]
cabin section [X] ||| chuancang qu [X]
section [X] ||| qu [X]
[X][X] had [X][X] [X] ||| [X][X] yi [X][X] [X]
had [X] ||| yi [X]
dropped into place [X] ||| hualuo [X]
. [X] |||  . [X]
```

Given the following input sentence

```
overhead oxygen masks in the cabin section
had dropped into place .
```

the Chinese translation by HPB is

```
toudingshangde yangqi chuancang qu de
mianzhao yi hualuo .
```

[2] https://code.google.com/p/berkeleyparser/
[3] http://www.statmt.org/moses/
[4] http://www.cis.upenn.edu/ ccb/ppdb/
[5] http://www.statmt.org/moses/download/sample-models.tgz

It is wrong ordered which leads to a totally wrong meaning. However, were a native Chinese speaker to read the translation, it is very hard to see if the sentence is correct or not. This is due to the fact that the left NT of the rule "X in the X" wrongly covers only "masks", not the "over head oxygen masks". This is a very common mistake in the HPB model.

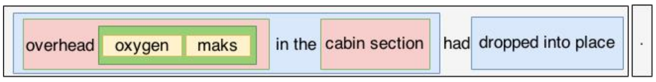


Figure 6: Authored With ProphetMT

However, with the following input generated by using ProphetMT, shown in Figure 6 by an native English speaker without any training, the correct translation can be generated using the same HPB model as:

```
chuancangqu de toudingshangde yangqi
mianzhao yi hualuo .
```

7. Discussion and Future Work

In this paper we describe ProphetMT, which is, to the best of our knowledge, the first syntactic SMT-driven CL authoring tool. ProphetMT provides both in-domain phrase-based and syntactic suggestions, which are compatible with an existing tree-based SMT system, to the user via auto-completion. By employing a novel shift-reduce-like scheme, users can naturally write from left-to-right while also parsing the composed parts and visualising the parsing results. With the help of the NodeBox component, the syntactic structure is rendered to the user in a simple format which does not require linguistic expertise to understand. Using the gold standard parsing results from the interface, a highly reliable SMT output can be generated which will reduce the post-editing efforts required later in the translation pipeline.

We hypothesise that ProphetMT will be suitable for monolingual CL writers of technical reports and manuals, patents, as well as contracts which are domain-specific and intented to be translated downstream. Furthermore, the tool is useful for more than just SMT — ProphetMT can also be helpful in maintaining consistency in writing styles across organizations, and in training beginners.

Future Work: In order to provide efficient auto-suggestions, we want NTRs which are relatively short, and "meaningful" to humans. The following improvements to the existing system can be implemented:

- Filtering the rules with syntactic parsing results as shown in Li et al. (2012) would be another way to prune the hierarchical phrase table.
- Dependency tree-to-string SMT model (Xie et al., 2011) as well as constituency tree-based model (Liu et al., 2006; Zhang et al., 2007) could also be interesting approaches for filtering out ungrammatical rules.
- Manual filtering may need to be performed for some domains, especially when ambiguity is costly.

The final evaluation of ProphetMT involves two different criteria:

- Measuring the composition time or the average edits as reported in the interactive machine translation tools (Foster et al., 2002; Alabau et al., 2014).
- Measuring the final translation quality relative to unaided composition.

The evaluation can be conducted by allowing human native speakers using ProphetMT to paraphrase test data.

8. References

Aho, A. V. (2003). *Compilers: Principles, Techniques and Tools (for Anna University), 2/e*. Pearson Education India.

Alabau, V., Buck, C., Carl, M., Casacuberta, F., Garcıa-Martınez, M., Germann, U., González-Rubio, J., Hill, R., Koehn, P., Leiva, L., et al. (2014). Casmacat: A computer-assisted translation workbench. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006). Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Du, J., Jiang, J., and Way, A. (2010). Facilitating translation using source language paraphrase lattices. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 420–429. Association for Computational Linguistics.

Foster, G., Langlais, P., and Lapalme, G. (2002). User-friendly text prediction for translators. In *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, jul. TT2 TransType2.

Gough, N. and Way, A. (2003). Controlled generation in example-based machine translation. In *MT Summit IX*. New Orleans, LO.

Gough, N. and Way, A. (2004). Example-based controlled translation. In *In Proceedings of the Ninth Workshop of the European Association for Machine Translation*. EAMT, Valetta, Malta.

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. Association for Computational Linguistics.

Li, J., Tu, Z., Zhou, G., and van Genabith, J. (2012). Using syntactic head information in hierarchical phrase-based translation. In *Proceedings of the Seventh Workshop on*

Statistical Machine Translation, pages 232–242. Association for Computational Linguistics.

Liu, Y., Liu, Q., and Lin, S. (2006). Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616. Association for Computational Linguistics.

Marti, J., Ahs, D., Lee, B., Falkena, J., Nelson, J., Kohlmeier, B., Liger, F., Pamarthi, R., Lerum, C., Mody, V., et al. (2010). User interface for machine aided authoring and translation, May 4. US Patent 7,711,546.

Mirkin, S., Venkatapathy, S., Dymetman, M., and Calapodescu, I. (2013). Sort: An interactive source-rewriting tool for improved translation. In *ACL (Conference System Demonstrations)*, pages 85–90. Citeseer.

Mitamura, T. (1999). Controlled language for multilingual machine translation. In *Proceedings of Machine Translation Summit VII, Singapore*, pages 46–52.

Nyberg, E., Mitamura, T., and Huijsen, W.-O. (2003). for authoring and translation. *Computers and Translation: A Translator's Guide*, 35:245.

O'Brien, S. (2003). Controlling controlled english. an analysis of several controlled language rule sets. *Proceedings of EAMT-CLAW*, 3:105–114.

O'Brien, S. (2005). Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation*, 19(1):37–58.

Power, R., Scott, D., and Hartley, A. (2003). Multilingual generation of controlled languages.

Stolcke, A. et al. (2002). Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Venkatapathy, S. and Mirkin, S. (2012). An smt-driven authoring tool. In *COLING (Demos)*, pages 459–466. Citeseer.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Xie, J., Mi, H., and Liu, Q. (2011). A novel dependency-to-string model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 216–226. Association for Computational Linguistics.

Zhang, M., Jiang, H., Aw, A., Sun, J., Li, S., and Tan, C. L. (2007). A tree-to-tree alignment-based model for statistical machine translation. *MT-Summit-07*, pages 535–542.