# Document Information as Side Constraints for Improved Neural Patent Translation

**Laura Jehl** jehl@cl.uni-heidelberg.de
Computational Linguistics, Heidelberg University, Germany

**Stefan Riezler** riezler@cl.uni-heidelberg.de
Computational Linguistics & IWR, Heidelberg University, Germany

**Abstract**

We investigate the usefulness of document information as side constraints for machine translation. We adapt two approaches to encoding this information as features for neural patent translation: As special tokens which are attached to the source sentence, and as tags which are attached to the source words. We found that sentence-attached features produced the same or better results as word-attached features. Both approaches produced significant gains of over 1% BLEU over the baseline on a German-English translation task, while sentence-attached features also produced significant gains of 0.7% BLEU on a Japanese-English task. We also describe a method to encode document information as additional phrase features for phrase-based translation, but did not find any improvements.

## 1 Introduction

Document information beyond the text is readily available in many data sets, but is rarely used when building translation systems. Such information could comprise the document's origin (time, place, author), its topic, or its connections to other documents. It exists, for example, in patents, textual content on the web, or e-commerce data. We aim to investigate the usefulness of document information as *side constraints* for translation. We use the term *side constraints* as it is used in Sennrich et al. (2016a) to mean information that is not present in a source string, but can influence translation choice in the target string.[1] For example, patents are assigned to a hierarchical classification system indicating their topic(s) in various degrees of granularity. Depending on the topic, different translation choices may be required. The correct choice will not always be apparent from the sentence context. By providing the classification information to the translation model, we enable the model to select the correct translation, given the constraints.

In this paper, we focus on patent translation. Patent translation lends itself particularly well to our endeavor since patent documents are annotated with different types of information, from hierarchical categorization to information about individual inventors. We are interested in seeing whether patent translation can be improved by this information and if so, which kind of information and which model integration is most useful.

Since 2014, neural machine translation (NMT) has become the state of the art in machine translation (Luong and Manning, 2015; Jean et al., 2015). We hypothesize that NMT is well-suited to the integration of document information. Since the model works on sentence representations, it can decide whether or not to pay attention to this information for each particular

[1] Sennrich et al. (2016a) explore politeness as a side constraint for translation.

translation in context. What is more, annotations could be highly correlated. For example, a patent document's subclass label contains its class label. Deep models are capable of learning these correlations.

Based on work by Kobus et al. (2016), this paper explores two ways of integrating patent document annotations as features into a neural machine translation system: (1) by attaching the annotations as special tokens to the source sentence, and (2) by attaching annotations as tags to each source word. Unlike our predecessors, we consider a setting where we have multiple annotation categories, and where each category can have more than one value at the same time. To our knowledge, our work is also the first to apply these methods to patent translation. We show that these features can positively influence translation choice of ambiguous words or phrases in a neural patent translation system. We also describe a method of integrating patent annotations into phrase-based machine translation (PBMT) as additional phrase features, but found that these features were unable to improve the model.

Related work is reviewed in Section 2. Section 3 describes the details of our approach. Using the experimental setup described in Section 4, we found that the simple approach of attaching annotations as additional tokens to the source sentence produced significant improvements – 0.7% BLEU for Japanese-English translation and 1% BLEU for German-English translation – in the right configuration. For German-English, similar improvements were gained by attaching the same information as tags to the source words. Detailed results are discussed in Section 5.

## 2 Related Work

### 2.1 Side Constraints

Our work is inspired by the work of Kobus et al. (2016) on multi-domain adaptation. This work uses the domain label as a side constraint for translation in a multi-domain setup: A combined NMT model is trained on subcorpora from different domains, and each training sentence is marked with its subcorpus information. Test data come from one of the known domains and are marked in the same way. We take the idea of using sentence-attached and word-attached source side features to represent side constraints from this paper and modify it to fit our scenario of multi-category, multi-valued patent annotations. Kobus et al. (2016) observed no improvements for sentence-attached features, but saw an improvement of 0.8% BLEU when testing on all but the largest subdomain.

Incorporating side constraints via *sentence-attached* features has also been applied in other work: Originally, this method was proposed by Sennrich et al. (2016a) to model politeness as a side constraint. Johnson et al. (2016) have used it to indicate the desired target language for multilingual NMT. Chu et al. (2017) apply it to neural domain adaptation in combination with fine tuning methods (Luong and Manning, 2015). Passing additional information to a neural network via *word-attached* features was first introduced by Collobert et al. (2011) as a way to add linguistic annotation for various NLP tasks using feed-forward and convolutional networks. Sennrich and Haddow (2016) transferred this idea to neural translation models. The word-attached features used by Kobus et al. (2016) were first presented by Crego et al. (2016), where they were used to encode case information.

The idea of leveraging document information as a side constraint for translation was recently investigated by Chen et al. (2016). They focus on integrating product category information for translation of product descriptions in e-commerce, and also apply their method to online lecture translation (Cettolo et al., 2012), where the lectures are annotated with topic keywords. They also experimented with attaching document information as an artificial token to the source sentence, but found no gains for this method. They then propose to integrate topic information on the target side by including it as an additional read-out layer in the decoder before the softmax layer. This method improved e-commerce data translation by 1.4% BLEU, lecture translation by 0.3% BLEU. For e-commerce translation, only product titles are used, for which it is likely that there is less context information available than would be for product descriptions.

We do not know of any prior work on using document information as side constraints for phrase-based machine translation. Niehues and Waibel (2010) and Bisazza et al. (2011) both modify the phrase table to include corpus (or in-domain/out-of-domain) identifiers, which they find beneficial for domain adaptation. However, they do not use more fine-grained information. For phrase-based patent translation, Wäschle and Riezler (2012b) use patent section labels to partition training data for multi-task learning, but do not look into the more fine-grained classification information.

### 2.2 Relation to Domain Adaptation

This work is related to the problem of domain adaptation in machine translation, which has been researched extensively for phrase-based translation, among others by Foster and Kuhn (2007); Axelrod et al. (2011); Matsoukas et al. (2009); Chen et al. (2013); Eidelman et al. (2012); Hewavitharana et al. (2013); Hasler et al. (2014), and is currently also being explored for neural machine translation (see Freitag and Al-Onaizan (2016); Zhang et al. (2016); Chen and Huang (2016); Chu et al. (2017); Wang et al. (2017); Chen et al. (2017) inter alia). There are two main scenarios for domain adaptation in the literature: In the first scenario, a translation model is adapted to a known, fixed target domain. A sample from the target domain is usually available. The aim of adaptation in this scenario is to make use of the in-domain sample to shift the model parameters to better match the target distribution. In a second scenario, called dynamic adaptation, the target domain is unknown and possibly shifting. No in-domain sample is available. Data from the target domain are only provided at test time, and their domain may change with each document. In this scenario, unsupervised methods have been used to infer a soft domain or topic attribution of the test context (document or sentence).

Our scenario is similar to dynamic adaptation, as we do not restrict our test data to one subdomain, but allow test data from any patent section or class. This setup precludes us from using the methods proposed for the first scenario, because they would require us to re-train the model for each document. However, unlike dynamic adaptation, in our case document information is provided as side constraints. As we are specifically interested in ways to utilize this information, we do not consider the unsupervised approaches to dynamic adaptation in this work. In future work, it could be informative to compare using human-assigned document information to inferred topic distributions, or to combine both sources of information.

## 3 Side Constraints for Patent Translation

We extract five types of side constraints from patent document annotations. The first three constraints reflect a patent's classification according to the hierarchical international patent classification system (IPC).[2] Patents are classified by their subject matter as belonging to one or more of 8 different sections (A-H). The sections are divided into 130 classes and 639 subclasses. The hierarchy branches out further, but we only use the top three hierarchy levels, which we call *IPC1* to *IPC3*. We also treat the name of the filing company (*COMP*) and the names of the inventors (*INV*) as side constraints. In this section, we describe how we integrate these side constraints into PBMT and NMT as phrase-, sentence-, and word-attached features. Since each patent can be assigned to more than one section, class and subclass, and be filed by more than one company or inventor, we are dealing with multiple feature categories which can have multiple values for each document.

[2] see `www.wipo.int/classifications/ipc/en/`

### 3.1 Phrase-based Machine Translation

In PBMT, side constraints can influence phrase selection. Given an annotated input document, it is our intuition that the model should prefer translations that have been extracted from documents with the same annotations. For example, the German-English phrase table contained both *impact plates* and *deflector plates* as translation candidates for the word *Prallplatten* (see Table 5, EXAMPLE 2, for the context). The German input document had been assigned to IPC section A. Of the two translation candidates, only *impact plates* had been seen in a document from IPC section A. This points to *impact plates* being the more suitable translation. We capture the degree of overlapping annotations between training and test document via additional phrase table features. At training time, the annotations of all documents containing a phrase pair are extracted. At test time, the intersection between the phrase pair annotations and the test document annotations is computed for each phrase. The actual features are computed by counting the overlapping annotations for each feature category (IPC section, class, and subclass; filing company and inventor). In total, five count-based features for the five patent specific feature categories were added to the standard PBMT model features.

### 3.2 Neural Machine Translation

**Sentence-attached features** Despite poor performance in previous work on domain adaptation, our first approach takes up the idea of integrating side constraints by attaching them as special tokens to the source sentence. On the one hand, this approach is simple and efficient, as it does not increase the number of model parameters. On the other hand, it allows the model to learn to pay attention to the beginning or end of sentence as needed. Since we are dealing with multiple values per category, we append each value to the sentence in alphabetical order. For example, if a patent document has been assigned to IPC sections B and C, the marked up input will look as follows:

<IPC1:B> <IPC1:C> *in die Matrix sind Verstärkungsfasern ( 5 , 6 , 7 ) eingebettet .*

Previous work has differed on where to attach the tag(s), leading us to run experiments for attaching at the front, back, or both front and back of a sentence.

**Word-attached features** Our second approach attaches side constraints as features to each input word, e.g.:

*in*|IPC1:B *die*|IPC1:B *Matrix*|IPC1:B …

Note that in our case, the same annotations will be attached to every word in one source document. In the sequence-to-sequence model without word-attached side constraints, the hidden encoder state $h^{(t)}$ at time $t$ is computed recursively as:

$$h^{(t)} = tanh(W(E_0 x_0^{(t)}) + Uh^{(t-1)})$$

where $E_0$ is the word embedding matrix, and $x_0^{(t)}$ is the $t$-th source word, represented as a one-hot vector. The model with word-attached features computes $h^{(t)}$ from a concatenation of the source word embedding and a vector representation $r_f$ of each word-attached feature $f$ as follows:

$$h^{(t)} = tanh(W([E_0 x_0^{(t)}, r_1, \ldots, r_F]) + Uh^{(t-1)})$$

where [ ] signifies vector concatenation and $r_1, \ldots, r_F$ are representations of each word-attached feature $f$.

There are different ways of computing $r_f$: In Kobus et al. (2016), the representation $r_f$ of feature $f$ for input $x_f^{(t)}$ at time $t$ is constructed as follows:

$$r_f^{(t)}[i] = \begin{cases} \frac{1}{|f|} & \text{if } x_f^{(t)} \text{ has the } i\text{-th value of } f \\ 0 & \text{otherwise} \end{cases}$$

where $|f|$ is the number of possible values of $f$ and $r_f \in \mathbb{R}^{|f|}$. Hence, they use a normalized sparse vector representation. This is unproblematic if $f$ only takes few values, but would become unwieldy for large $|f|$.

As we want to be able to handle features with many possible values, we would like to use a dense representation $r_f$. Following Sennrich and Haddow (2016)'s approach for adding linguistic features to the NMT input, we want to construct feature representations using separate embedding matrices for each feature. The matrices are learned during training and feature representations are computed via a lookup layer. The hidden state $h^{(t)}$ of the encoder RNN at time $t$ is then computed as:

$$h^{(t)} = tanh(W([E_0 x_0^{(t)}, E_1 x_1^{(t)}, \ldots, E_F x_F^{(t)}]) + Uh^{(t-1)})$$

where [ ] signifies vector concatenation, $E_0 x_0^{(t)}$ computes the source word embedding, and $E_1 \ldots E_F$ are separate embedding matrices for each feature type, with $x_1 \ldots x_F$ encoding each feature's value as a one-hot vector.

This approach assumes that each feature only takes one value and representations can be computed efficiently by embedding lookup. Since patent documents can have more than one value for the same annotation category, our setup does not meet this assumption. We solve the problem by looking up the embeddings for all values of the same feature $f$ in the same embedding matrix $E_f$, and then summing over embeddings belonging to the same feature. The representation $r_f$ for feature $f$ is then computed as:

$$r_f = \sum_{i=1}^{K_f} E_f x_{f,i}^{(t)}$$

where $x_{f,i}^{(t)}$ is a one-hot vector encoding the $i$-th value of feature $f$ at source position $t$, and $K_f$ is a hyperparameter to be set by the user, which determines the maximum number of values each feature can take. We pick this value by looking at the distribution of the number of values for each feature in the training documents, and select a cutoff value if less than 5% of training documents have more values for the same feature. For documents with more than $K_f$ annotations, a subset of the annotations is sampled. For documents with fewer than $K_f$ annotations for feature $f$, empty values are marked by an extra token.

We select this approach for comparison with the sentence-attached features, because it also operates on the source-side and does not necessarily require increasing the model parameters. See our system description in Section 4 for details. The advantage of word-attached features is that this method makes combining multiple side constraints easier, as we cannot attach very long sequences of special tokens to the beginning and end of the sentence. The concatenated embeddings will also allow the model to learn correlations between annotations.

## 4 Experiment Setup

### 4.1 Data

We ran experiments on Japanese-English and German-English patent translation. We trained Japanese-English models on the NTCIR-7 Patent Translation training set (Utiyama and Isahara,

| Data set | Origin | # sentence pairs | # documents |
|---|---|---|---|
| train | NTCIR-7 train | 1,798,571 | 51,040 |
| dev | NTCIR-8 pat-dev-2006-2007 | 2,000 | 115 |
| devtest | NTCIR-8 Test Intrinsic | 1,251 | 114 |
| test | NTCIR-9 Test Intrinsic | 2,001 | 417 |

Table 1: Data sets used in Japanese-English translation

| Data set | Origin | # sentence pairs | # documents |
|---|---|---|---|
| train | PatTR abstracts before 2008 | 694,609 | 280,009 |
| dev | PatTR abstracts since 2008 | 2,000 | 899 |
| devtest | PatTR abstracts since 2008 | 2,001 | 864 |
| test (filtered) | PatTR abstracts since 2008 | 1,716 | 724 |

Table 2: Data sets used in German-English translation

2007). We used the NTCIR-8 development and test sets for development, and the NTCIR-9 intrinsic evaluation set for testing. For German-English, we used the abstracts section of the PatTR corpus (Wäschle and Riezler, 2012a). Patents published before 2008 were used for training. Development, devtest and test data of about 2,000 sentences each were randomly selected from the remaining patents. Sentences from the same document were always assigned to the same set. We applied a length-ratio based filter to the test set prior to evaluation to filter out noise from automatic sentence alignment. Tables 1 and 2 contain information about the data sets. Japanese data was segmented using MeCab[3]. All English data was tokenized and true-cased using the Moses toolkit[4]. German data was tokenized, lowercased, and compounds were split, also using Moses tools.

### 4.2 Translation Systems

We used the Nematus NMT system[5] (Sennrich et al., 2017) to train an attentional encoder-decoder network (Bahdanau et al., 2015). The model parameters were optimized with ADADELTA (Zeiler, 2012), using a maximum sentence length of 80 and a minibatch size of 80. We trained a subword model using BPE (Sennrich et al., 2016b) with 29,500 merge operations. We used 500-dimensional word embeddings and set hidden layer size to 1024. For the sentence-attached features, we experiment with IPC section (*IPC1*) and class (*IPC2*) labels, appending them at the front, back, or front and back of the source sentence. For the word-attached features we used IPC section and class labels (*IPC1/IPC2*) together. The maximum number of values per feature for *IPC1* and *IPC2* were set to 2 and 3, the embedding sizes were set to 5 and 20. In order to avoid improvements from merely increasing the number of parameters, we used 475-dimensional source word embeddings when adding word-attached features. The concatenated embedding vectors then have the same length (500) as the original word embedding vectors. We trained all NMT models using early stopping based on training cost on heldout data with a patience of 10. Results are reported on the final model.

For the PBMT experiments, we trained and tested a hierarchical PBMT model using cdec (Dyer et al., 2010). The baseline used 21 built-in dense features. A 5-gram target-side language

[3] taku910.github.io/mecab/

[4] github.com/moses-smt/mosesdecoder

[5] github.com/EdinburghNLP/nematus

|  | BLEU↑ | TER↓ |
|---|---|---|
| PBMT Baseline | 27.2 | 57.8 |
| *5 phrase features* | 27.2 | 58.3 |
| NMT Baseline | 36.9 | 49.3 |
| *sentence, IPC1, front* | 37.4 | 49.1 |
| *sentence, IPC1, back* | 37.5* | 48.5* |
| *sentence, IPC1, front/back* | 37.6* | 48.3* |
| *sentence, IPC2, front* | 37.2 | 49 |
| *sentence, COMP, front* | 37 | 48.9 |
| *word, IPC1/IPC2* | 37.2 | 49.2 |

Table 3: Japanese-English translation results.

|  | BLEU↑ | TER↓ |
|---|---|---|
| PBMT Baseline | 41.7 | 45 |
| *5 phrase features* | 41.7 | 45.2 |
| NMT Baseline | 42.5 | 47.2 |
| *sentence, IPC1, front* | 43.5* | 45.6* |
| *sentence, IPC1, back* | 43.2* | 46.3* |
| *sentence, IPC1, front/back* | 42.7 | 46.9 |
| *sentence, IPC2, front/back* | 43.9* | 45.3* |
| *word, IPC1/IPC2* | 43.5* | 46.3* |

Table 4: German-English translation results.

model was built with `lmplz` (Heafield et al., 2013). Feature weights were trained with `dtrain` (Simianer et al., 2012) for 15 epochs. Results are reported on the final epoch. All five annotations categories (IPC section, class, and subclass, company, inventor) are used to compute 5 phrase level features.

We report BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) on tokenized output, as computed by `multeval` (Clark et al., 2011).

## 5 Results

Table 3 shows results for Japanese-English patent translation, German-English results are shown in Table 4. For both language pairs, adding side constraints to PBMT did not improve the baseline. Results which improve significantly over the NMT baseline at $p \leq 0.05$ are marked with an asterisk*.

For Japanese-English, switching to NMT improved scores strikingly (+9.7% BLEU, -8.4% TER), probably due to NMT's superiority at handling word order differences in long patent sentences. Attaching IPC section labels (*IPC1*) as special tokens at the *front and back* of the sentence produced a small, significant, improvement over the NMT baseline (+0.7% BLEU, -1% TER), as did attaching them at the *back* of the sentence. Attaching IPC section labels (*IPC1*), class labels (*IPC2*) or company names (*COMP*) at the *front* of the source sentence did not significantly improve the baseline, nor did the word-attached features.

For German-English, the performance gap between PBMT and NMT was much narrower at 0.9% BLEU, probably due to more similar word order. We even see a reversed ranking for

TER (+2.2%). Unlike Japanese-English, attaching IPC section labels (*IPC1*) as special tokens at the *front* of the sentence improved significantly over the NMT baseline (+1% BLEU, -1.6% TER), while attaching them at the *front and back* did not produce a significant improvement. Interestingly, attaching IPC class labels (*IPC2*) at the *front and back* also produced a significant improvement (+1.4% BLEU, -1.9% TER). Word-attached features also produced a significant improvement (+1% BLEU, -0.9% TER), but did not outperform sentence-level features. These results differ from previous work, where sentence-attached domain or topic labels produced no gains. We also tried to combine word-attached and sentence-attached features, but did not see additional gains.

Due to time constraints we were only able to evaluate all sentence-attachment variations for both language pairs for the *IPC1* feature. For Japanese-English, attaching features at the *back* and *front/back* was more successful than attaching them at the *front* of the sentence. For German-English, attachment at the *front* produced better BLEU and TER scores then attachment at the *back*, and *front/back* was worst. This observation invites speculation that there could be a connection between the more beneficial attachment location and the word order of the source language leading the model to pay more attention to the front or back of the sentence, but further experiments would be necessary to confirm or dismiss this speculation. For now, we can conclude from our experiments that there is no general recommendation on which attachment location is best.

Table 5 shows example sentences from the German-English test set and their translations by different models. In EXAMPLE 1, the NMT model with side constraints correctly translated the German word “Kupplung” as *clutch*, which was incorrectly translated as *coupling* by the PBMT and NMT baseline. The correct phrase translation for “elektrischen Maschine”, *electric machine*, was also only selected by the model with side constraints. In EXAMPLE 2, the correct translation *impact plates* for German “Prallplatten” was produced by all NMT models. However, the word “Wasserschleiers” was only translated correctly as *water curtain* by the model with side constraints. It was passed through the decoder in PBMT and omitted entirely by the NMT baseline. The NMT model with side constraints also selected the correct translation *steam cabin* for German “Dampfkabine”, where PBMT produced *cubicle* and the NMT baseline produced *steam booth*.

## 6 Conclusion

In this paper, we have investigated methods for using document information as side constraints for phrase-based and neural translation models for patent translation. Document information was incorporated into the model as phrase-, word-, or sentence-attached features. Contrary to previous work, we have looked at incorporating multiple annotation categories with multiple values. For phrase-based machine translation, our features based on annotation overlap between test documents and phrase context were not helpful. For neural machine translation, attaching patent annotations as special tokens to the source sentence – a method which was unsuccessful in previous work – improved German-English translation by over 1% BLEU, and Japanese-English translation by 0.7% BLEU. Word-attached features also produced improvements of 1% BLEU for German-English patent translation, but did not improve Japanese-English translation significantly. Overall, the results indicate that document information can improve patent translation and that neural machine translation is well-suited to integrating this kind of information. However, choosing the right configuration requires some experimentation.

## Acknowledgements

This work was supported by DFG Research Grant RI 2221/1-2 “Weakly Supervised Learning of Cross-Lingual Systems”. We thank the anonymous reviewers for their insightful comments.

| | |
|---|---|
| Example 1 | |
| Source | (. . . ) mit einer **Kupplung** ( 8 ) , die den Verbrennungsmotor ( 2 ) auswählbar vollständig mit der **elektrischen Maschine** ( 10 ) selektiv verbindet (. . . ) |
| PBMT baseline | (. . . ) with a coupling ( 8 ) , the internal combustion engine ( 2 ) can be completely selectively connects with the electrical machine ( 10 ) (. . . ) |
| NMT baseline | (. . . ) comprising a coupling ( 8 ) which can be selectively connected to the electric motor ( 10 ) (. . . ) |
| +IPC1 front | (. . . ) comprising a **clutch** ( 8 ) which selectively connects the internal combustion engine ( 2 ) to the **electric machine** ( 10 ) (. . . ) |
| Reference | (. . . ) having a **clutch** ( 8 ) which selectively connects the internal combustion engine ( 2 ) completely to the **electric machine** ( 10 ) (. . . ) |
| Example 2 | |
| Source | Sie weist (. . . ) **Prallplatten** ( 531 ) auf zur Erzeugung eines flächigen **Wasserschleiers** ( 25 ) zumindest im oberen Bereich der **Dampfkabine** (. . . ) |
| PBMT baseline | it has (. . . ) deflector plates ( 531 ) for generating a flat wasserschleiers ( 25 ) at least in the upper region of the cubicle (. . . ) |
| NMT baseline | (. . . ) **impact plates** ( 531 ) , at least in the upper region of the steam booth (. . . ) |
| +IPC1 front | (. . . ) it has (. . . ) **impact plates** ( 531 ) for producing a flat **water curtain** ( 25 ) at least in the upper region of the **steam cabin** (. . . ) |
| Reference | (. . . ) said steam cabin comprises (. . . ) **impact plates** ( 531 ) for producing a flat **water curtain** ( 25 ) at least in the upper area of the **steam cabin** . (. . . ) |

Table 5: Examples for German-English translation: We compare output from PBMT and NMT baselines to the NMT model with sentence-attached IPC section features (+IPC1 front). **Boldfaced** portions highlight correct translations. Incorrect translations are underlined.

## References

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362, Edinburgh, United Kingdom.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations* , San Diego, CA, USA.

Bisazza, A., Ruiz, N., and Federico, M. (2011). Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proceedings of the International Workshop on Spoken Language Translation*, San Francisco, CA, USA.

Cettolo, M., Girardi, C., and Federico, M. (2012). WIT$^3$: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16$^{th}$ Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chen, B., Cherry, C., Foster, G., and Larkin, S. (2017). Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46, Vancouver, Canada.

Chen, B. and Huang, F. (2016). Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 314–323, Berlin, Germany.

Chen, B., Kuhn, R., and Foster, G. F. (2013). Vector Space Model for Adaptation in Statistical Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1285–1293, Sofia, Bulgaria.

Chen, W., Matusov, E., Khadivi, S., and Peter, J. (2016). Guided Alignment Training for Topic-Aware Neural Machine Translation. *arXiv preprint arXiv:1607.01628*.

Chu, C., Dabre, R., and Kurohashi, S. (2017). An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 176–181, Portland, Oregon.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Crego, J., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., Akhanov, E., Brunelle, P., Coquard, A., Deng, Y., et al. (2016). SYSTRAN's Pure Neural Machine Translation Systems. *arXiv preprint arXiv:1610.05540*.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). `cdec`: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 115–119, Jeju Island, Korea.

Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT)*, Prague, Czech Republic.

Freitag, M. and Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Hasler, E., Haddow, B., and Koehn, P. (2014). Dynamic Topic Adaptation for SMT using Distributional Profiles. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 445–456, Baltimore, Maryland, USA.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Hewavitharana, S., Mehay, D., Ananthakrishnan, S., and Natarajan, P. (2013). Incremental topic-based translation model adaptation for conversational spoken language translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 697–701, Sofia, Bulgaria.

Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015). Montreal Neural Machine Translation Systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Kobus, C., Crego, J., and Senellart, J. (2016). Domain Control for Neural Machine Translation. *arXiv preprint arXiv:1612.06140*.

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Singapore.

Niehues, J. and Waibel, A. (2010). Domain adaptation in statistical machine translation using factored translation models. In *Proceedings of the Annual Conference of the European Association or Machine Translation*, Saint-Rapha"el, France.

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

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Sennrich, R. and Haddow, B. (2016). Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, CA, USA.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany.

Simianer, P., Riezler, S., and Dyer, C. (2012). Joint Feature Selection in Distributed Stochastic Learning for Large-scale Discriminative Training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, United States.

Utiyama, M. and Isahara, H. (2007). A Japanese-English patent parallel corpus. *Proceedings of the MT summit XI*, pages 475–482.

Wang, R., Utiyama, M., Liu, L., Chen, K., and Sumita, E. (2017). Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark.

Wäschle, K. and Riezler, S. (2012a). Analyzing Parallelism and Domain Similarities in the MAREC Patent Corpus. *Multidisciplinary Information Retrieval*, pages 12–27.

Wäschle, K. and Riezler, S. (2012b). Structural and topical dimensions in multi-task patent translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 818–828, Avignon, France.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, J., Li, L., Way, A., and Liu, Q. (2016). Topic-informed neural machine translation. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1807–1817, Osaka, Japan.