Visual-Interactive Neural Machine Translation

Category: Research

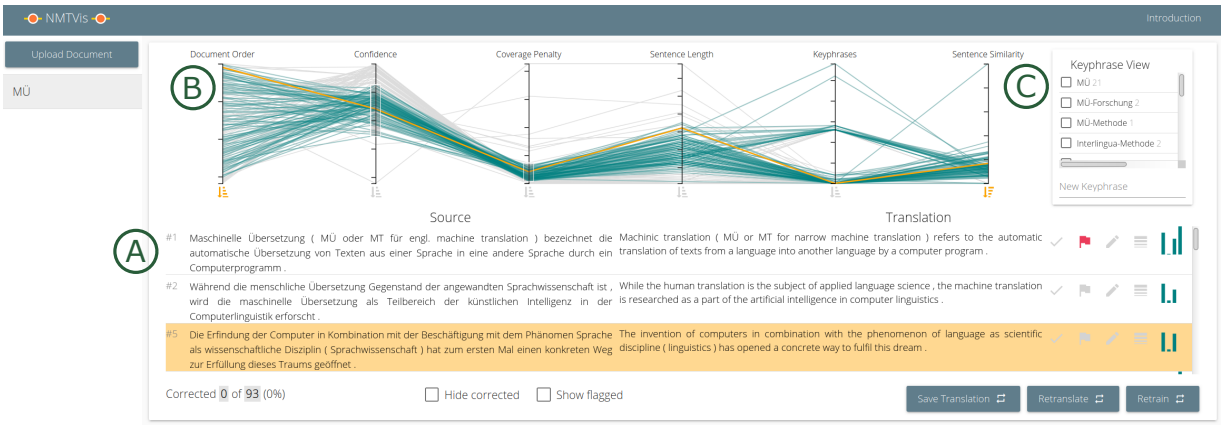

Figure 1: The main view of our neural machine translation system: (A) Document View with sentences of the document for the current filtering settings, (B) Metrics View with sentences of the filtering result highlighted, and (C) Keyphrase View with a set of rare words that may be mistranslated. The Document View initially contains all sentences automatically translated with the NMT model. After filtering with the Metrics View and Keyphrase View, a smaller selection of sentences is shown. Each entry in the Document View provides information about metrics, the correction state, and functionality for modification (on the right side next to each sentence). The Metrics View represents each sentence as one path and shows values for different metrics (e.g., correlation, coverage penalty, sentence length). Green paths correspond to sentences of the current filtering. One sentence is highlighted (yellow) in both the Metrics View and the Document View.

Abstract

We introduce a novel visual analytics approach for analyzing, understanding, and correcting neural machine translation. Our system supports users in automatically translating documents using neural machine translation and identifying and correcting possible erroneous translations. User corrections can then be used to fine-tune the neural machine translation model and automatically improve the whole document. While translation results of neural machine translation can be impressive, there are still many challenges such as over- and under-translation, domain-specific terminology, and handling long sentences, making it necessary for users to verify translation results; our system aims at supporting users in this task. Our visual analytics approach combines several visualization techniques in an interactive system. A parallel coordinates plot with multiple metrics related to translation quality can be used to find, filter, and select translations that might contain errors. An interactive beam search visualization and graph visualization for attention weights can be used for post-editing and understanding machine-generated translations. The machine translation model is updated from user corrections to improve the translation quality of the whole document. We designed our approach for an LSTM-based translation model and extended it to also include the Transformer architecture. We show for representative examples possible mistranslations and how to use our system to deal with them. A user study revealed that many participants favor such a system over manual text-based translation, especially for translating large documents.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Visual analytics; Human-centered computing—Visualization—Visualization systems and tools; Computing methodologies—Artificial intelligence—Natural language processing—Machine translation

1 Introduction

Machine learning and especially deep learning are popular and rapidly growing fields in many research areas. The results created with machine learning models are often impressive but sometimes still problematic. Currently, much research is performed to better understand, explain, and interact with these models. In this context, visualization and visual analytics methods are suitable and more and more often used to explore different aspects of these models. Available techniques for visual analytics in deep learning were examined by Hohman et al. [16]. While there is a large amount of work available for explainability in computer vision, less work exists for machine translation.

As it becomes increasingly important to communicate in different languages, and since information should be available for a huge range of people from different countries, many texts have to be translated. Doing this manually takes much effort. Nowadays, online translation systems like Google Translate [13] or deepL [10] support humans in translating texts. However, the translations generated that way are often not as expected or like someone familiar with both languages might translate them. It may also not express someone's translation style or use the correct terminology of a specific domain or for some occasion. Often, more background knowledge about the text is required to translate documents appropriately.

With the introduction of deep learning methods, the translation quality of machine translation models has improved considerably in the last years. However, there are still difficulties that need to be addressed. Common problems of neural machine translation (NMT) models are, for instance, over- and under-translation [35] when words are translated repeatedly or not at all. Handling rare words [20], which might be available in specific documents, and long sentences, are also issues. Domain adaption [20] is another challenge; especially documents from specific domains such as medicine, law, or science require high-quality translations [7]. As many NMT models are trained on general data sets, their translation

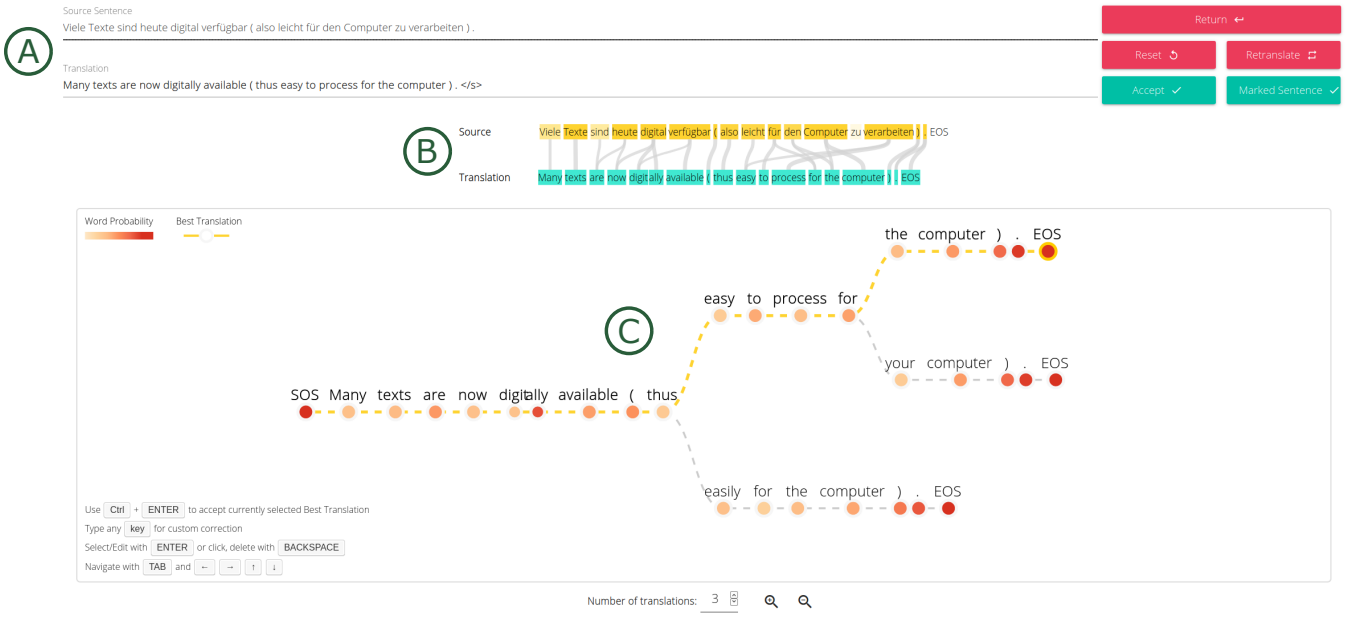

Figure 2: The detailed view for a selected sentence consists of the Sentence View (A), the Attention View (B), and the Beam Search View (C). The Sentence View allows text-based modifications of the translation. The Attention View shows the attention weights (represented by the lines connecting source words with their translation) for the translation. The Beam Search View provides an interactive visualization that shows different translation possibilities and allows exploration and correction of the translation. All three areas are linked.

performance is worse for domain-specific texts.

If high-quality translations for large texts are required, it is insufficient to use machine translation models alone. These models are computationally efficient and able to translate large documents with low time effort, but they may create erroneous or inappropriate translations. Humans are very slow compared to these models, but they can detect and correct mistranslations when familiar with the languages and the domain terminology. In a visual analytics system, both of these capabilities can be combined. Such a system should provide the translations from an NMT model and possibilities for users to visually explore translation results to find mistranslated sentences, correct them, and steer the machine learning model.

We have developed a visual analytics approach to reach the goals outlined above. First, our system performs automatic translation of a whole, possibly large, document and shows the result in the Document View (Figure 1). Users can then explore and modify the document on different views [28] (Figure 2) to improve translations and use these corrections to fine-tune the NMT model. We support different NMT architectures and use both an LSTM-based and a Transformer architecture.

So far, visual analytics systems for deep learning were mostly available for computer vision, some text-related areas, focusing on smaller parts of machine translation [22, 27] or intended for domain experts to gain insight into the models or to debug them [32, 33]. This work contributes to visualization research by introducing the application domain of NMT using a user-oriented visual analytics approach. In our system, we employ different visualization techniques adapted for usage with NMT. Our parallel coordinates plot (Figure 1 (B)) supports the visualization of different metrics related to text quality. The interaction techniques in our graph-based visualization for attention (Figure 2 (B)) and tree-based visualization for beam search (Figure 2 (C)) are specifically designed for text exploration and modification. They have a strong coupling to the underlying model. Furthermore, our system has a fast feedback loop and allows interaction in real-time. We demonstrate our system's features in a video and will provide the source code of our system with the published paper.

2 Related Work

This section first discusses visualization and visual analytics approaches for language translation in general and then visual analytics of deep learning for text. Afterward, we provide an overview of work that combines both areas in the context of NMT.

Many visualization techniques and visual analytics systems exist for text; see Kucher and Kerren [21] for an overview. However, there is little work on exploring and modifying translation results. An interactive system to explore and correct translations was introduced by Albrecht et al. [1]. While the translation was created by machine translation, their system did not use deep learning. Lattice structures with an uncertainty that can be used to visualize machine translation were used by Collins et al. [9]. They created a lattice structure from beam search where the path for the best translation result is highlighted and can be corrected. We also use visualization for beam search, but ours is based on a tree structure.

Recently, much research was done to visualize deep learning models to understand them better. Multiple surveys [6, 12, 16, 23, 45] are available that give summaries of existing visual analytics systems. It is noticeable that not much work exists related to text-based domains. One of the few examples is RNN-Vis [24], a visual analytics system designed to understand and compare models for natural language processing by considering hidden state units. Karpathy et al. [18] explore the prediction of Long Short-Term Memory (LSTM) models by visualizing activations on text. Heatmaps are used by Hermann et al. [15] in order to visualize attention for machine-reading tasks. To explore the training process and to better understand how the network is learning, RNNbow [4] can be used to visualize the gradient flow during backpropagation training in Recurrent Neural Networks (RNNs).

While the previous systems support the analysis of deep learning models for text domains in general, approaches exist to specifically explore and understand NMT. The first who introduced visualizations for attention were Bahdanau et al. [2]; they showed the contri-

bution of source words to translated words within a sentence, using an attention weight matrix. Later, Rikters et al. [27] introduced multiple ways to visualize attention and implemented exploration of a whole document. They visualize attention weights with a matrix and a graph-based visualization connecting source words and translated words by lines whose thickness represents the attention weight. Bar charts give an overview for a whole document for multiple attention-based metrics that are supposed to correlate with the translation quality. Interactive ordering of these metrics and sentence selection is possible. However, it is difficult for large documents to compare the different metrics as each bar chart is horizontally too large to be entirely shown on a display. The only connection between different bar charts is that the bars for the currently selected sentence are highlighted. Our system also uses such a metrics approach, but instead of using bar charts, a parallel coordinates plot was chosen for better scalability, interaction, and filtering.

An interactive visualization approach for beam search is provided by Lee et al. [22]. The interaction techniques supported by their tree structure are quite limited. It is possible to expand the structure and to change attention weights. However, it is not possible to add unknown words, and no sub-word units are considered. Furthermore, the exploration is limited to single sentences instead of a whole document.

With LSTMVis, Strobelt et al. [33] introduced a system to explore LSTM networks by showing hidden state dynamics. Among other application areas, their approach is also suitable for NMT. While our approach is rather intended for end-users, LSTMVis has the goal of debugging models by researchers and machine learning developers. With Seq2Seq-Vis, Strobelt et al. [32] present a system that uses an attention view similar to ours, and they also provide an interactive beam search visualization. However, their system is designed to translate single sentences, and no model adaption is possible for improved translation quality. Their system was designed for debugging and for gaining insight into the models.

Since there are different architectures available for generating translations [43], specific visualization approaches may be required. Often, LSTM- based architectures are used. Recently, the Transformer architecture [36] gained popularity; Vig [37, 38] visually explores their self-attention layers and Rikters et al. [26] extended their previous approach for debugging document to Transformer-based systems.

All these systems provide different, possibly interactive, visualizations. However, their goal is rather to debug NMT models instead of supporting users in translating entire documents, or they are limited to small aspects of the model. Additionally, they are usually designed for one specific translation model. None of these approaches provide extended interaction techniques for beam search or interactive approaches to iteratively improve the translation quality of a whole document.

3 Visual Analytics Approach

Our visual analytics approach allows the automatic translation, exploration, and correction of documents. Its components can be split into multiple parts. First, a document is automatically translated from one language into another one, then mistranslated sentences in the document are identified by users, and individual sentences can be explored and corrected. Finally, a model can be fine-tuned and the document retranslated.

Our approach has a strong link to machine data processing and follows the visual analytics process presented by Keim et al. [19]. We use visualizations for different aspects of NMT models, and users can interact with the provided information.

3.1 Requirements

For the development of our system, we followed the nested model by Munzner [25]. The main focus was on the outer parts of the model, including identifying domain issues, feature implementation design, and visualization and interaction implementation. Additionally, we used a similar process as Sedlmair et al. [29], especially focusing on the core phases. Design decisions were made in close cooperation with deep learning and NMT experts, who are also co-authors of this paper. The visual analytics system was implemented in a formative process that included these experts. Our system went through an iterative development that included multiple meetings with our domain experts. Together, we identified the requirements listed in Table 1. After implementing the basic prototype of the system, we demonstrated it to further domain experts. At a later stage, we performed a small user study with experts for visualization and machine translation. For our current prototype, we added recommended functionality from these experts.

Table 1: Requirements for our visual analytics system and their implementations in our approach.

R1	**Automatic translation** – A document is translated automatically by an NMT model.
R2	**Overview** – The user can see the whole document as a list of all source sentences and their translations (Figure 1 (A)). Additionally, an overview of the translation quality is provided in the Metrics View that reveals statistics about different metrics encoded as a parallel coordinates plot (Figure 1 (B)) showing an overall quality distribution.
R3	**Find, filter, and select relevant sentences** – Interaction in the parallel coordinates allows filtering according to different metrics and selecting specific sentences. It is also possible to select one sentence and order the other sentences of the document by similarity to verify for similar sentences if they contain similar errors. Additionally, our Keyphrase View (Figure 1 (C)) supports selecting sentences containing specific keywords that might be domain-specific and rarely used in general documents.
R4	**Visualize and modify sentences** – For each sentence, a beam search and attention visualization (Figure 2) can be used to interactively explore and adapt the translation result in order to correct erroneous sentences and explore how a translation failed. It is also possible to explore alternative translations.
R5	**Update model and translation** – The model can be fine-tuned using the user inputs from translation corrections; this is especially useful for domain adaption. Afterward, the document is retranslated with the updated model in order to improve the translation result (the result is visualized similar to Figure 9).
R6	**Generalizability and extensibility** – While we initially designed our visualization system for one translation model, we soon noticed that our approach should handle data from other translation models as well. Therefore, our approach should be easily adaptable for new models to cope with the dynamic development of new deep learning architectures. Our general translation and correction process is held quite agnostic to be applied on a variety of models. Only model-specific visualizations have limitations, need to be adapted or exchanged when using a different translation architecture.
R7	**Target group** – The target group for our system should be quite broad and include professional translators or students who need to translate documents. However, it should also be able to be used by other people interested in correcting and possibly better understanding the results of automated translation.

3.2 Neural Machine Translation

The goal of machine translation is to translate a sequence of words from a source language into a sequence of words in a target language. Different approaches exist to achieve this goal [34, 44].

Usually, neural networks for machine translation are based on an encoder-decoder architecture. The encoder is responsible for transforming the source sequence into a fixed-length representation known as a context vector. Based on the context vector, the decoder generates an output sequence where each element is then

used to generate a probability distribution over the target vocabulary. These probabilities are then used to determine the target sequence; a common method to achieve this uses beam search decoding [14].

Although different NMT models vary in their architecture, the previously described encoder-decoder design should apply to a wide range of architectures and new approaches that may be developed in the future (R6). In this work, we explored an LSTM architecture with attention and extended our approach to include the Transformer architecture, thus verifying its ability to generalize.

One of the first neural network architectures for machine translation consists of two RNNs with LSTM units [5]. To handle long sentences, the attention mechanism for NMT [2] was introduced. It allows sequence-to-sequence models to attend to different parts of the source sequence while predicting the next element of the target sequence by giving the decoder access to the encoder's weighted hidden states. During decoding, the hidden states of the encoder together with the hidden state of the decoder for the current step are used to compute the attention scores. Finally, the context vector for the current step is computed as a sum of the encoder hidden states, weighted by the attention scores. The attention weights can be easily visualized and used to explain why a neural network model predicted a certain output. Furthermore, attention weights can be understood as a soft alignment between source and target sequences. For each translated word, the weight distribution over the source sequence signifies which source words were most important for predicting this target word. The Transformer architecture was recently introduced by Vaswani et al. [36] and gained much popularity. It uses a more complex attention mechanism with multi-head attention layers; especially, self-attentions play an important role in the translation process. We verify its applicability to our approach and visualize only the part of the attention information that showed an alignment between source and target sentences comparable to the LSTM model.

3.3 Exploration of Documents

After uploading a document to our system, it is translated by an NMT model (R1). The main view of our approach then shows information about the whole document (R2). This includes a list of all sentences in the Document View (Figure 1 (A)) and an overview of the translation quality in the Metrics View (Figure 1 (B)). Using the Metrics View and Keyphrase View (Figure 1 (C)), sentences can be filtered to detect possible mistranslated sentences that can be flagged by the user (R3). Once a mistranslated sentence is found, it is also possible to filter for sentences containing similar errors (R3).

Metrics View

In the *Metrics View*, a parallel coordinates plot (Figure 1 (B)) is used to detect possible mistranslated sentences by filtering sentences according to different metrics (R3). For instance, it is possible to find sentences that have low translation confidence.

Multiple metrics exist that are relevant to identify translations with low quality; we use the following metrics in our approach:

- **Confidence:** A metric that considers attention distribution for input and output tokens; it was suggested by Rikters et al. [27]. Here, a higher value is usually better.
- **Coverage Penalty:** This metric by Wu et al. [42] can be used to detect sentences where words did not get enough attention. Here, a lower value is usually better.
- **Sentence length:** The sentence length (the number of words in a source sentence) can be used to filter very short or long sentences. For example, long sentences might be more likely to contain errors.
- **Keyphrases:** This metric can be used to filter for sentences containing domain-specific words. As these words are rare in the training data, the initial translation of sentences containing them is likely erroneous. The values used for this metric are the number of occurrences of keyphrases in a sentence weighted by the frequency of the keyphrases in the whole document.
- **Sentence similarity:** Optionally, for a given sentence, the similarity to all other sentences can be determined using cosine similarity. This helps to find sentences with similar errors to a detected mistranslated sentence.
- **Document index:** The document index allows the user to sort sentences according to their original order in the document, which can be especially important for correcting translations where the context of sentences is relevant. Furthermore, this metric might also show trends like consecutive sentences with low confidence.

In contrast to Rikters et al. [27], who use bar charts to visualize different metrics, we chose a parallel coordinates plot [17]. Each sentence can be mapped to one line in such a plot, and different metrics can be easily compared. These plots are useful for an overview of different metrics and to detect outliers and trends. Interactions with the metrics, such as highlighting lines or choosing filtering ranges, are supported. It can be expected that sentences filtered for both, low confidence and high coverage penalty, are more likely to be poorly translated than sentences falling into only one of these categories.

Keyphrase View

It is possible to search for sentences according to keyphrases by selecting them in the *Keyphrase View* (Figure 1 (C)) (R3). This can be visualized as shown in Figure 4. Keyphrases are domain-specific words and were not often included in the training data used for our model. As the model has not enough knowledge on how to deal with these words, it is important to verify if the respective sentences were translated correctly. In addition to automatically determined keyphrases, users can manually specify further keyphrases for sentence filtering.

Document View

A list of all the source sentences in a document and a list of their translations are shown in the *Document View* (Figure 1 (A)) (R2). Each entry in this list can be marked as correct or flagged (Figure 4) for later correction. A small histogram shows an overview of the previously mentioned metrics. If a sentence is modified, either through user-correction or retranslation by the fine-tuned model, changes in the sentences are highlighted (Figure 9). Both the Metrics View and the Keyphrase View are connected via brushing and linking [39] to allow filtering for sentences that are likely to be mistranslated and should be examined and possibly corrected. Additionally, sentences can be sorted into a list by similarity to a user-selected reference sentence. In this list, sentences can be selected for further exploration and correction in more detailed sentence-based views.

3.4 Exploration and Correction of Sentences

After filtering and selection, a sentence can be further analysed with the Sentence, Attention, and Beam Search Views (Figure 2) and subsequent correction (R4). These views are shown simultaneously to allow interactive exploration and modification of translations.

Note, on the sentence level, we use subword units to handle the problem of rare words, which often occurs in domain-specific documents, and to avoid unknown words. We use the Byte Pair Encoding (BPE) method proposed by Sennrich et al. [31] for compressing text by recursively joining frequent pairs of characters into new subwords. This means, instead of choosing whole words to build the source and target vocabulary, words are split into subword units consisting of possibly multiple characters. This method reduces model size, complexity, and training time. Additionally, the model

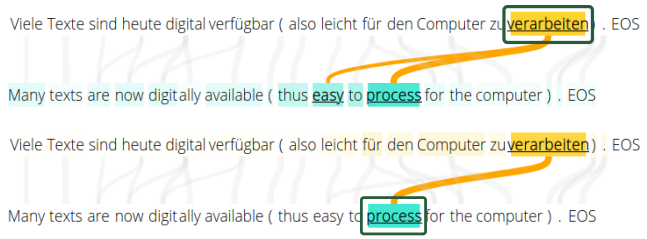

Figure 3: Attention visualization: (top) when hovering a source word (here: 'verarbeiten') translated words influenced by the source are highlighted and (bottom) when hovering a translated word (here: 'process') source words that influence the translation are highlighted according to attention weights.

can handle unknown words by splitting them into their subword units. As these subword units are known beforehand, they do not require the introduction of an "unknown" token for translation. Thus, we can adapt the NMT model to any new domain, including those with vocabulary not seen at training time.

Sentence View

Similar to common translation systems, the *Sentence View* (Figure 2 (A)) shows the source sentence and the current translation. It is possible to manually modify the translation, which in turn updates the content in the other sentence-based views. After adding a new word in the text area, the translation with the highest score is used for the remainder of the sentence. This supports a quick text-based modification of a translation without explicit use of visualizations.

Attention View

The *Attention View* depends on the underlying NMT model. It is intended to visualize the relationship between words of the source sentence and the current translation as a weighted graph (Figure 2 (B)); such a technique was also used by Strobelt et al. [32]. Both source and translated words are represented by nodes; links between such words show the attention weights encoded by the thickness of the connecting lines (we use a predefined threshold to hide lines for very low attention). These weights correlate with the importance of source words for the translated words. Hovering over a source word highlights connecting lines to translated words starting at this word. In addition, the translated words are highlighted by transparency according to the attention weights (Figure 3 top). While this shows how a source word contributes to the translation, it is also possible to show for translated words how source words contribute to the translation (Figure 3 bottom). This interactive visualization supports users in understanding how translations are generated from the source sentence words. On the one hand, such a visualization helps gain insight into the NMT model, and, on the other hand, detects issues in generated translations. The links between source sentence and translation can be explored to identify anomalies such as under- or over-translation. Missing attention weights can be an indication for under-translation and links to multiple translated words for over-translation. In our case study in Section 4, examples of these cases are presented. While this technique specifically employs information of the attention-based LSTM model, we use it in an adapted form for the Transformer architecture (see Section 4.4). A visualization more tailored to Transformers, also including self-attention and attention scores from multiple decoder layers, could provide additional information. Further models may need different visualizations for a generalized use of our approach, employing model-specific information.

Beam Search View

While the Attention View can be used to identify positions with mistranslations, the *Beam Search View* supports users in interactively modifying and correcting translations. The Beam Search View visualizes multiple translations created by the beam search decoding as a hierarchical structure (see Figure 2 (C)). This interactive visualization can be used for post-editing the translations.

The simplest way of predicting a target sequence is greedy decoding, where at every time step, the token with the highest output probability is chosen as the next predictied token and fed to the decoder in the next step. This is an efficient, straightforward way of generating an output sequence. However, another translation may be better overall, despite having lower probabilities for the first words. Beam search decoding [14] is a compromise between exhaustive search and greedy decoding, often used for generating the final translation. A fixed number (k) of hypotheses is considered at each timestep. For each hypothesis considered, the NMT model outputs a probability distribution over the target vocabulary for the next token. These hypotheses are sorted by the probability of the latest token, and up to k hypotheses remain in the beam. Hypotheses ending with the End-of-Sequence (EOS) token are filtered out and put into the result set. Once k hypotheses are in the result set, the beam search stops, and the final hypotheses are ranked according to a scoring function that depends on the attention weights and the sentence length.

For visualization, we use a similar approach as Strobelt et al. [32], and Lee et al. [22]: a tree structure reflects the inherently hierarchical nature of the beam search decoding. This way, translation hypotheses starting with the same prefixes are merged into one branch of this hierarchical structure. The root node of each translation is associated with a Start-of-Sequence (SOS) token and all leaf nodes with an End-of-Sequence (EOS) token. Compared to the visualization of a list of different suggested translations, showing a tree is more compact, and it is easier to recognize where commonalities of different translation variants lie.

Each term of the translation is visualized by a circle that represents the actual node and a corresponding label. The color of a circle is mapped to the word's output probability. This supports users in identifying areas with a lower probability that might require further exploration. It can be seen as uncertainty for the prediction of words. In our visualization, we differentiate between nodes that represent subwords and whole words. Continuous lines connect subwords and nodes are placed closer together to form a unit. In contrast, the connections to whole words are represented by dashed lines.

The beam search visualization can be used to navigate within a translation and edit it (Figures 7 and 8). The interaction can be performed either mouse-based or keyboard-based; the latter is more efficient for fast post-editing. The view supports standard panning-and-zooming techniques that are especially needed to explore long sentences as they do not fit common displays. For navigation within the tree, arrow keys can be used to move through a sentence, or nodes can be selected by mouse cursor. If the translation of the current node's child node is not satisfying, the node can be expanded to show suggestions for correction. If the user selects a suggested word, the beam search runs with a lexical prefix constraint, and the tree structure gets updated. If the suggested words are not suitable, a custom correction can be performed by typing an arbitrary word that better fits. The number of suggested translations is initially set to three and can be increased by adapting the beam size. Increasing this value may create better translations and provides more alternative translations. However, the higher the value is, the more information has to be shown in the visualization. By hovering and selecting elements in this view, corresponding elements of the Attention View and Sentence View are shown for reference.

Figure 4: Main view of the system: the Document View shows some flagged sentences for correction. Additionally, the keyphrase filter (top right) is active: all sentences containing the keyphrase 'MÜ' are shown in the Metrics and Document Views. It is visible that 'MÜ' is never correctly translated to 'MT'.

3.5 Model Fine-tuning and Retranslation

After correcting the translation of multiple sentences, the user corrections can be used to fine-tune the NMT model and automatically improve the translation of the not yet verified sentences (R5). This approach can be applied repeatedly to improve the document's translation quality, especially for domain-specific texts.

Documents often belong to a specific domain, e.g., legal, medical, or scientific. Each domain has specific terminology, and one word may even refer to different concepts in different domains. As such, the ability of NMT models to handle different types of domains is an important research topic. Domain adaption refers to techniques allowing NMT models trained on general training data, also called out-of-domain, to adapt to domain-specific documents, called in-domain. This is useful since there may be an abundant amount of general training data, but domain-specific data may be rare. Since NMT models need a large amount of training data to achieve good translation quality, the out-of-domain data can be used to train a baseline model. The model can then be adapted using the in-domain data (R5), which typically contains a smaller number of sentences: in our system, we use the user-corrected sentences. This mitigates the problem of training an NMT model in a low-resource scenario where little data exists for a given domain. In our approach, we continue training for the in-domain data in a reduced way by freezing certain model weights (for the LSTM-based model, both decoder and the LSTM layers of the encoder are trained; for the Transformer, only the decoder is trained).

4 Case Study

As a typical use case, we take the German Wikipedia article for *machine translation* (*Maschinelle Übersetzung*) [41] as a document for translation into English. In the following, we show how to use our system to improve the translation quality of the document. Please see our accompanying video for a demonstration with the Transformer model. The examples in the following were created with both the LSTM and Transformer models. We trained our models on a general data set: the German-to-English data set from the 2016 ACL Conference on Machine Translation (WMT'16) [3] shared news translation task. This is a popular data set for NMT, used, for instance, by Denkowski and Neubig [11] and Sennrich et al. [30].

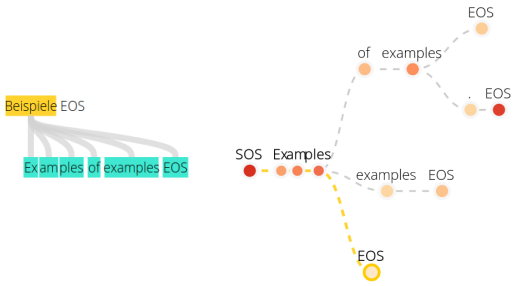

Figure 5: Example of over-translation: 'Examples' is placed twice as translation for the German word 'Beispiele'. The Beam Search View (right) shows possible alternative translations. However, only increasing the beam size to four shows the translation we would have expected.

4.1 Exploration of Documents

After uploading a document (R1), we have a look at the parallel coordinates plot (R2) for our initial translations and the list of keyphrases in order to detect possible mistranslations (R3). In the Keyphrase View, we notice the domain-specific term 'MÜ' occurring very often. This term is the German abbreviation for 'machine translation' and should therefore be translated as 'MT'. However, none of the translations use the correct term (Figure 4). Additionally, one could select and verify sentences with low confidence or with a high coverage penalty. Here, we especially notice the under-translation of some sentences. After verifying a translation in the Document View, users can decide if they are correct (R2). If the users do not agree with the translation, they can set a flag (Figure 4) to modify the translation later or switch to the sentence-based views to correct it (R4).

4.2 Exploration and Correction of Sentences

After setting flags for multiple sentences (Figure 4), or the decision to explore or modify a sentence, a more detailed view for each sentence can be shown to explore and improve their translations interactively (Figure 2) (R4).

Over-translation is a common issue of NMT [20]. In the Attention View, it is possible to see what went wrong by identifying where the attention weights connect the source and destination words.

For both models, we notice some cases for very short sentences. Figure 5 shows for the German heading 'Beispiele' (en: 'Examples'), a translation that uses the translated word multiple times. Also, the suggested alternatives use this term more than once. Only after increasing the beam size to four, the correct translation is visible, which can then be selected as the correction.

More often, only parts of a sentence are translated, and important words are not considered in our document. Such **under-translation** is shown in Figure 6. In the first example, only the beginning of the sentence is translated, and it is visible that the remaining nodes have almost zero attention. In the second example, the German term 'zweisprachigen' (en: 'bilingual') is skipped in the translation. While this part of the translation is missing, the translated sentence is still correct and fluid; it might be difficult to detect such an error without such attention visualizations.

An example of a wrong translation containing a **keyphrase** is visualized in Figure 7. Here, it is also shown that using the beam search visualization, it is possible to select an alternative translation interactively starting from the position where the first error occurs. The beam search provides possible alternative translations, but it is possible to manually type what the user believes should be the next term. Here, we enter the correct translation manually. The beam search visualization automatically updates in real-time according to the correction.

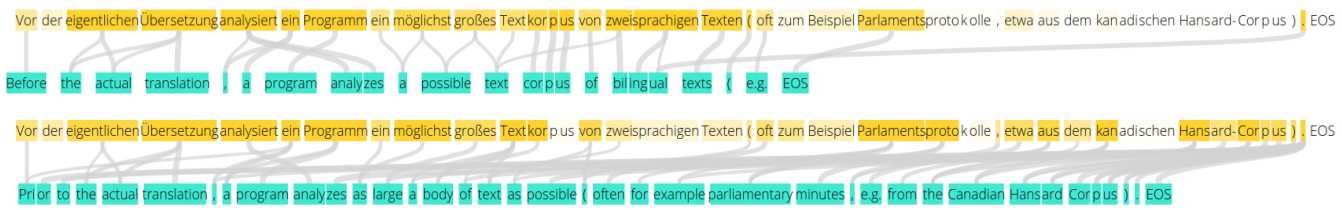

Figure 6: Example of under-translation shown in the Attention View: (top) for the LSTM model the end of the sentence is not translated; attention weights are very low for this part of the sentence. (Bottom) for the Transformer architecture the term 'zweisprachigen' (en: 'bilingual') is not translated; attention weights are very low for this term.

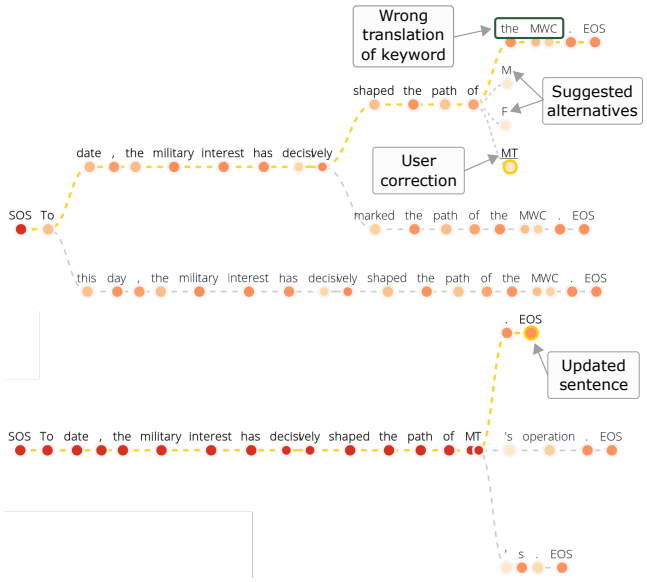

Figure 7: Example of a mistranslated sentence containing the keyphrase 'MÜ' shown as beam search visualization: (top) suggested translation, suggested alternatives and custom correction; (bottom) updated translation tree for corrected keyword with new suggestions for continuing the sentence after the custom change.

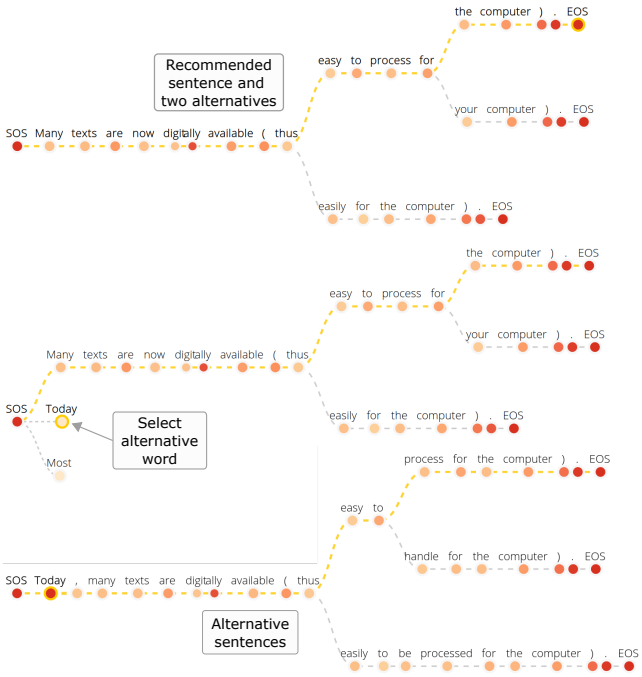

Figure 8: Correctly translated sentence is changed to another correct translation. 'SOS' is selected to show alternative beginnings for the sentence. After choosing an alternative the remaining sentence gets updated by another correct translation.

Finally, it is also possible to change sentences without mistakes. Sometimes, sentences are correctly translated, but different words or sentence structures are used as the current user would prefer for the context of a sentence or to express someone's own style (Figure 8). Again, it is possible to explore and select alternative words or sentences with the Beam Search View. If we wanted to start the sentence with a different word, an alternative could be selected, and the remaining sentence would get updated accordingly.

After correcting and accepting multiple translation corrections, the Document View shows how a translation was changed (Figure 9).

4.3 Model Fine-tuning and Retranslation

After users corrected multiple sentences, they can choose to retrain the current model for not yet accepted sentences (R5). The model is then fine-tuned using the corrected sentences by the user. This is usually a small number of sentences. Afterward, the system translates the uncorrected sentences to improve translation quality as the model adapted from the corrected sentences. Since our document contains 29 times the keyphrase 'MÜ' that is wrongly translated, we retrained our model after correcting only a few (less than 5) of these terms to 'MT'. After retranslation, the Document View shows the difference of the translations compared to before. For both the LSTM and the Transformer model, all or almost all occurrences of 'MÜ' are now correctly translated. The user can look at the changes and accept translations or continue with iteratively improving sentences and fine-tuning the model.

4.4 Architecture-specific Observations

We initially designed our approach for the use with an LSTM-based model with an attention mechanism. Since other architectures exist to translate documents, we also adapted it and tested its usefulness for the current state-of-the-art Transformer architecture [36] (R6). This architecture is also attention-based, and we analyzed how well it fits our interactive visualization approach. The general workflow of our system can be used in the same way as the model we initially developed it for: the Document and Metrics Views can be used to identify sentences for further investigation, and sentences can be updated using the Sentence and Beam Search View. The main difference between the Transformer model concerning our approach is the attention mechanism that influences the Attention View and some calculated metric values.

The Transformer architecture uses multiple layers with multiple self-attention heads instead of just attentions between encoder and

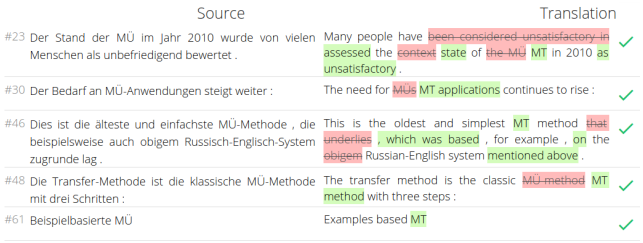

Figure 9: Document View showing corrected translations and changes to the initial machine-generated translations.

decoder. There are approaches for the visualization of this more complex attention mechanism [37, 38]. The attention values for Transformers could, for example, show different linguistic characteristics for different attention heads [8]. However, including this into our system would make our approach more complex and not useful for end-users (R7) with little knowledge about this architecture. As a simple workaround to apply our visualization, we discard the self-attention and only use the decoder attention. We explored the influence of decoder attention values from different layers, averaged across all attention heads. Similar to Rikters et al. [26], we noticed that averaging attention from all layers is not meaningful since almost all source words are connected to all target words. Using one of the first layers showed similar results. For the final layer, a better alignment could be seen; however, the last token of the source word received too much attention compared to other words. Instead, using the second last layer showed a similar alignment between source and target words as it is available for the LSTM model. Therefore, we use this as a compromise for the use in our Attention View and for calculation of metric values.

Since there are different approaches and architectures developed for NMT, we could incorporate them as well (R6). Some might provide better support in gaining insights into the model and offer different visualization and interaction capabilities. For others, new ways for visualization will have to be investigated.

5 User Study

We conducted an early user study during the development of our approach to evaluate our system's concept. We used a prototype with an LSTM translation model. The system had the same views as described before but limited features. A group of anonymous visualization and machine learning experts were invited to test our system online for general aspects related to visualization, interaction, and usefulness. Our goal was to make sure that we considered aspects relevant from both the visualization and the machine translation perspective in our system and to improve our approach. The user study was questionnaire-based to evaluate the effectiveness of the system, understandability of visualizations, and usability of interaction techniques. A 7-point Likert scale was used. In this study, the German Wikipedia article for *autonomous driving* (*Autonomes Fahren*) [40] was available to all participants. This allowed the participants to explore the phenomena we showed previously. The participants claimed to have good English (mean = 5.1, std. dev. = 0.8) and very good German (mean = 6.2, std. dev. = 1.7) knowledge. While the visualization experts claimed to have rather low knowledge about machine learning (mean: 2.5), the machine learning experts similarly indicated lower knowledge for visualization (mean: 3).

First, participants were introduced to the system with a short overview of the features. Then, they could explore the system freely with no time restriction. Afterward, they were asked to participate in a survey regarding the usefulness of our system and its design choices. Additionally, there were free text sections for further feedback. We recruited 11 voluntary participants from our university (six experts on visualization and five for language processing).

Table 2: Ratings from our user study for each evaluated view on a 7-point Likert scale; mean and standard deviation values are provided.

View	Effectiveness	Visualization	Interaction
Metrics View	5.9 (1.1)	6.8 (0.4)	6.1 (0.7)
Keyphrase View	4.4 (1.6)	6.5 (1.2)	6.3 (1.1)
Beam Search View	5.6 (1.5)	6 (1.3)	4.5 (1.8)
Attention View	5.6 (0.8)	6.2 (1.2)	5.9 (0.9)

The general effectiveness of translating a large document containing more than 100 sentences with our approach was rated high (mean = 5.6, std. dev. = 1.0) compared to a small document containing up to 20 sentences (mean = 4.5, std. dev. = 1.6). The results for effectiveness, ease of understanding and intuitiveness of visualizations, and ease of interaction are given in Table 2. The ratings for the visualizations were high for all views. Best rated was the Metrics View that additionally had the lowest standard deviation. As not all our user study participants were visualization experts, we noticed that non-experts could also manage to understand and work with parallel coordinate plots. We conclude that our design choice for the visualization of metrics was appropriate. The ratings for interaction were also very high, but there was more variation. Especially the interaction for beam search was rated comparatively low and had the highest standard deviation; two language processing participants ranked it very low (1 and 2) and two (one from each participant group) very high (7). This variation might be the result of the learning curve being different for different participant groups. Since we conducted the user study, we have also improved the interaction in this view. For effectiveness, the Keyphrase View had the lowest rating. We believe the reason is that participants were not able to detect enough mistranslated sentences with this view. However, this might be due to our document provided and may differ for other documents containing more domain-specific vocabulary as we showed in our case study.

In addition, we asked users for general feedback on our approach. Especially the Metrics View received positive feedback. Participants mentioned that it is useful for quickly detecting mistranslations through brushing and linking. For the Beam Search View, one participant noted that the alternatives provided would speed up the correction of translations. For one participant, the Attention View was useful in showing the differences in the sentence structure of different languages. Negative feedback was mostly related to interaction and specific features; some participants suggested new features. Multiple participants noted that the exploration and correction of long sentences are challenging in the Beam Search View as the size of the viewport is limited. Furthermore, a feature to delete individual words and functionality for freezing areas was suggested. From the remaining feedback, we already included, for example, an undo function for the sentence views. Also, to find sentences that might contain similar errors, one participant recommended showing sentences similar to a selected sentence, and we added a respective metric. Additionally, it was mentioned that confidence scores could be shown in the document list next to each sentence and not only in the Metrics View. This would be helpful to quickly examine the confidence value even if the document is sorted by a different metric (e.g., document order); small histograms were added next to each sentence as a quick quality overview.

6 Discussion and Future Work

To conclude, we present a visual analytics approach for exploring, understanding, and correcting translations created by NMT. Our approach supports users in translating large domain-specific documents with interactive visualizations in different views, allows sentence

correction in real-time and model adaption.

Our qualitative user study results showed that our visual analytics system was rated positively regarding effectiveness, interpretability of visualizations, and ease of interaction. The participants mastered the translation process well with our selected visualizations. Especially, our choice of parallel coordinate plots for visualization of multiple metrics and the related interaction techniques for brushing and linking were rated positively. Our approach had a clear preference for translating large documents compared to a traditional text-based approach. Right now, users have to use metrics to decide with which sentence they will start correcting the translations. More research has to be done for better automatic detection of mistranslated sentences. For example, an additional machine learning model could be trained with sentences that were already identified as wrong translations.

We believe our system is useful for people who have to deal with large documents and could use the features of interactive sentence correction and domain adaption. Comparing the use of our approach for LSTM and the Transfomer architecture showed almost no difference; for both, we could successfully interactively improve the translation quality of documents and see model-specific information. We argue that our general translation and visualization process can also be used with further models, while in such cases, some visualization views might need limited adaptation.

References

[1] J. Albrecht, R. Hwa, and G. E. Marai. The Chinese Room: Visualization and interaction to understand and correct ambiguous machine translation. *Computer Graphics Forum*, 28(3):1047–1054, 2009.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[3] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pp. 131–198. Association for Computational Linguistics, 8 2016.

[4] D. Cashman, G. Patterson, A. Mosca, and R. Chang. RNNbow: Visualizing learning via backpropagation gradients in recurrent neural networks. In *Workshop on Visual Analytics for Deep Learning (VADL)*, 2017.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014.

[6] J. Choo and S. Liu. Visual analytics for explainable deep learning. *IEEE Computer Graphics and Applications*, 38(4):84–92, Jul 2018.

[7] C. Chu and R. Wang. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1304–1319. Association for Computational Linguistics, 2018.

[8] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286. Association for Computational Linguistics, Florence, Italy, Aug. 2019. doi: 10.18653/v1/W19-4828

[9] C. Collins, S. Carpendale, and G. Penn. Visualization of uncertainty in lattices to support decision-making. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization (EuroVis 2007)*, pp. 51–58, 2007.

[10] DeepL. DeepL Translator. `https://www.deepl.com/translator`, 2021.

[11] M. Denkowski and G. Neubig. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 18–27. Association for Computational Linguistics, 2017.

[12] R. Garcia, A. C. Telea, B. C. da Silva, J. Tørresen, and J. L. D. Comba. A task-and-technique centered survey on visual analytics for deep learning model engineering. *Computers & Graphics*, 77:30–49, 2018.

[13] Google. Google Translate. `https://translate.google.com`, 2021.

[14] A. Graves. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012.

[15] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., *Advances in Neural Information Processing Systems 28*, pp. 1693–1701. Curran Associates, Inc., 2015.

[16] F. M. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[17] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, Aug 1985. doi: 10.1007/BF01898350

[18] A. Karpathy, J. Johnson, and F. Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

[19] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In *Visual data mining*, pp. 76–90. Springer, 2008.

[20] P. Koehn and R. Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39. Association for Computational Linguistics, 2017.

[21] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 117–121, 2015.

[22] J. Lee, J.-H. Shin, and J.-S. Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 121–126. Association for Computational Linguistics, 2017.

[23] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48 – 56, 2017.

[24] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24, 2017.

[25] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[26] M. Rikters. Debugging neural machine translations. *arXiv preprint arXiv:1808.02733*, 2018.

[27] M. Rikters, M. Fishel, and O. Bojar. Visualizing neural machine translation attention and confidence. *The Prague Bulletin of Mathematical Linguistics*, 109(1):39–50, 2017.

[28] J. C. Roberts. State of the art: Coordinated multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pp. 61–71, 2007.

[29] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012. doi: 10.1109/TVCG.2012.213

[30] R. Sennrich, B. Haddow, and A. Birch. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pp. 371–376. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/w16-2323

[31] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725. Association for Computational Linguistics, 2016.

[32] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2seq-vis : A visual debugging tool for sequence-to-sequence

models. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[33] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018.

[34] Z. Tan, S. Wang, Z. Yang, G. Chen, X. Huang, M. Sun, and Y. Liu. Neural machine translation: A review of methods, resources, and tools, 2020.

[35] Z. Tu, Y. Liu, L. Shang, X. Liu, and H. Li. Neural machine translation with reconstruction. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3097–3103, 2017.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *ArXiv e-prints*, June 2017.

[37] J. Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42. Association for Computational Linguistics, Florence, Italy, July 2019. doi: 10.18653/v1/P19-3007

[38] J. Vig. Visualizing attention in transformerbased language models. *arXiv preprint arXiv:1904.02679*, 2019.

[39] M. O. Ward. *Linking and Brushing*, pp. 1623–1626. Springer US, Boston, MA, 2009. doi: 10.1007/978-0-387-39940-9_1129

[40] Wikipedia. Autonomes fahren — wikipedia, die freie enzyklopädie, 2018.

[41] Wikipedia. Maschinelle übersetzung — wikipedia, die freie enzyklopädie, 2021.

[42] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv e-prints*, Sept. 2016.

[43] S. Yang, Y. Wang, and X. Chu. A survey of deep learning techniques for neural machine translation. *arXiv preprint arXiv:2002.07526*, 2020.

[44] S. Yang, Y. Wang, and X. Chu. A survey of deep learning techniques for neural machine translation. 2020.

[45] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, pp. 1–34, 2020.