Statistical Machine Translation for Depassivizing German Part-of-speech Sequences

Benjamin Gottesman
Acrolinx
Friedrichstraße 100
10117 Berlin, Germany
ben.gottesman@acrolinx.com

Abstract

We aim to use statistical machine translation technology to correct grammar errors and style issues in monolingual text. Here, as a feasibility test, we focus on depassivization in German and we abstract from surface forms to parts of speech. Our results are not yet satisfactory but yield useful insights into directions for improvement.

1 Introduction

There exist software applications that identify errors in text and, in some instances, automatically generate possible corrections. However, they are not yet sophisticated enough to correct most grammar errors and style issues. We aim to train instances of the statistical machine translation (SMT) system Moses (Koehn et al, 2007) to perform monolingual reformulations aimed at correcting specific grammar and style issues. In other words, the system will 'translate' from error-filled text to correct text in the same language. If successful, this approach could be used to extend the capacity of existing text-checking software to generate corrections.

This work is at an early stage. What we present here is a feasibility test that is limited to a specific language and style issue: avoiding passive voice in German. Furthermore, we abstract from surface forms to parts of speech (POSes), thereby admittedly glossing over some complications, as we will show.

2 Related Work

This is related to earlier work on statistical post-editing. Dugast et al (2007), for example, trained Moses on a parallel corpus that paired the outputs of SYSTRAN, a rule-based machine translation (RBMT) system, with gold-standard human translations of the same input sentences. That is, they trained an SMT system to correct RBMT errors. Our training data, in contrast, pairs human-written sentences containing errors with human-edited versions of the same sentences in which the errors have been corrected.

This is also related to work in which POS information is used in SMT. Popović and Ney (2007) and Genzel (2010), for example, perform POS-based reordering on source-language sentences in order to make them more like the target language and thereby reduce the amount of reordering that must be performed by the SMT system. Stated more generally, they use POS information in a preprocessing step that makes the SMT task easier.

We, in contrast, provide the POSes as input to the SMT system itself, which makes our work more similar to work in the area of factored translation models. Koehn and Hoang (2007) presented these models and described experiments in which they were used as a means of annotating SMT training data with POSes and other lexical information. We do not technically use factored models, though, as we train our SMT systems on the POSes alone rather than on surface forms annotated with POSes.

3 Experiment Method and Results

Concisely, our method is as follows.

- check German text segments using Acrolinx
- manually correct issues identified by Acrolinx flags
- select segment pairs (before and after editing) in which the only edit was a depassivization
- convert segments to POS-tag sequences
- where two segment pairs are duplicates at POS-tag level, discard one
- partition segments into training/tuning/test sets
- train and tune SMT depassivizer and apply it to test set
- evaluate test output automatically and manually

The following subsections describe the method in more detail.

3.1 Data

We started from approximately 77,000 German text segments from the OPUS corpus (Tiedemann, 2012). This consists of technical documentation of the OpenOffice office productivity software suite. We checked these segments using Acrolinx, a commercial text-checking software product that flags spelling and grammar errors as well as style issues (as described by Bredenkamp et al (2000)). We then had a human editor edit the segments in response to the flags. The editor was not permitted to perform any edit that was not in response to a specific flag, but was permitted to ignore false flags or other flags for which there was no useful edit. For each edited segment, the editor noted which flag type(s) prompted the edit(s). Thus, we can train an SMT system to correct one specific type of error by selecting as training data just those segments containing relevant edits. Table 1 shows the number of edits for the most common flag types.

Flag type	# edited segments	# isolated* edits
Avoid passives	1411	571
Avoid ambiguous words	611	291
Avoid parentheses	554	393
Avoid more than two prepositional phrases	504	140
Use digits	347	178
Avoid pronouns with unclear referent	340	90
Avoid verbosity	331	121
...		

* *isolated* means there were no edits in the given segment other than for this flag type

Table 1: Data analysis: Edit count by flag type

The most common of all is *Avoid passives*, a flag that reflects the stylistic dispreference for passive voice in, for example, German technical writing (tekom, 2011). We focus on this flag type because of the amount of data and because its correction patterns are largely systematic yet not implemented by existing text-checking software. Thus, we shall try to train Moses to depassivize German sentences. We use only the isolated edits in order to avoid confusing the system with unrelated edits. The available data is thus 571 German OpenOffice text segments, before and after depassivization by a human editor, with no other edits. We use a POS tagger to convert the text segments to sequences of POS tags. After removing some duplicates among the segments, we partition the remainder arbitrarily into training, tuning, and test sets of 517, 20, and 10 POS-tag sequence pairs, respectively.

3.2 Common depassivization patterns

Let us digress for a moment from discussion of our experimental methodology to look at common depassivization patterns, as this will provide context to our analysis of the behaviour of our translation systems in the following subsection.

Based on inspection of a sampling of the edits performed by the human editor in response to *Avoid passives* flags, there is one canonical pattern for depassivizing German sentences and a second pattern, less common but still occurring repeatedly, that we refer to as the *verb-swap* pattern.

In a **canonical depassivization**, as illustrated in figure 1 (in English for the convenience of non-German-speaking readers),

- the subject noun phrase becomes an object

`NP1-subj` is `V-PP` → `NP2-subj V-finite NP1-obj`
e.g. ‘The apple is eaten.’ → ‘The man eats the apple.’

Figure 1: Canonical depassivization pattern

noun phrase (`NP1-subj` becomes `NP1-obj`),

- the verb is changed from passive to active voice, which typically involves dropping the auxiliary verb (*is* or a related form) and changing the full verb from participle form to finite (`V-PP` becomes `V-finite`), and
- a new subject noun phrase (`NP2-subj`) is introduced.

The third point is problematic for automatic depassivization. The new subject typically does not appear in the original sentence (except sometimes in a prepositional phrase); deciding its identity requires context and world knowledge. The best we can reasonably hope for from an automatic system (that operates at surface level) is to insert a ‘dummy’ subject: ‘X eats the apple’. By operating at the POS level, we bypass this issue.

Also, at the POS level, the first point is not necessarily detectable, in which case the pattern appears to consist of only two parts: the transformation of the verb and the introduction of a noun phrase.

`NP1-subj` is `V1-PP` → `NP1-subj V2-finite`
e.g. ‘The image is shown.’ → ‘The image appears.’

Figure 2: Verb-swap depassivization pattern

In the **verb-swap pattern** (figure 2), the transitive verb in passive voice is replaced by a semantically related intransitive verb in active voice.

The decision of when to use this pattern and the choice of which verb to introduce are both lexical – they depend semantically on the original verb. Working at the POS level thus simultaneously complicates matters, by removing information required for deciding whether to use this pattern, and simplifies them, by freeing the system from having to select a specific replacement verb.

3.3 Preliminary Results

Using the data described in section 3.1, we train two Moses systems: one standard phrase-based and one tree-based.

Since passives in German often involve long-distance dependencies, tree-based SMT is intuitively more promising for this task.

Table 2 gives the BLEU scores (Papineni et al, 2002) achieved by the respective systems on our test set. They suggest that the tree-based system is indeed slightly better.

System	BLEU
standard	72.57
tree-based	73.56

Table 2: BLEU scores achieved by our two systems

However, manual analysis reveals that the BLEU score difference is misleading and that both result sets are equally bad. The problem is that the system applies parts of the depassivization patterns independently of each other, and independently of whether there is a passive in a given clause.

In figure 3, for example, we see an illustration of test item #3, in which the standard phrase-based system performs only half of the canonical pattern: it correctly translates the infinitival passive verb form `VVPP VAINF` to an active infinitive verb `VVINF`, but it fails to insert the missing subject. (We reiterate that the system input, output, and reference are the POS sequences; surface forms are shown for the reader’s convenience.) The tree-based system produces the exact same result for this item.

The input string of test item #4 (figure 4), meanwhile, consists of two clauses, only the second of which contains a passive. The first clause should thus not be modified, but both systems (which, again, produce the exact same output string) perform half of the canonical depassivization, inserting a noun phrase. On the second clause, the systems perfectly perform the canonical depassivization, but, as we see in figure 4, the standard phrase-based system appears to be performing it as two independent changes. One change is the deletion of the verb participle `VVPP` and the other consists of the replacement of the finite auxiliary verb `VAFIN` with the finite full verb `VVFIN` and the insertion of a noun phrase. The tree-based system similarly performs these as two independent changes.

Both systems successfully apply the verb-swap pattern to item #1 (figure 5), producing output iden-

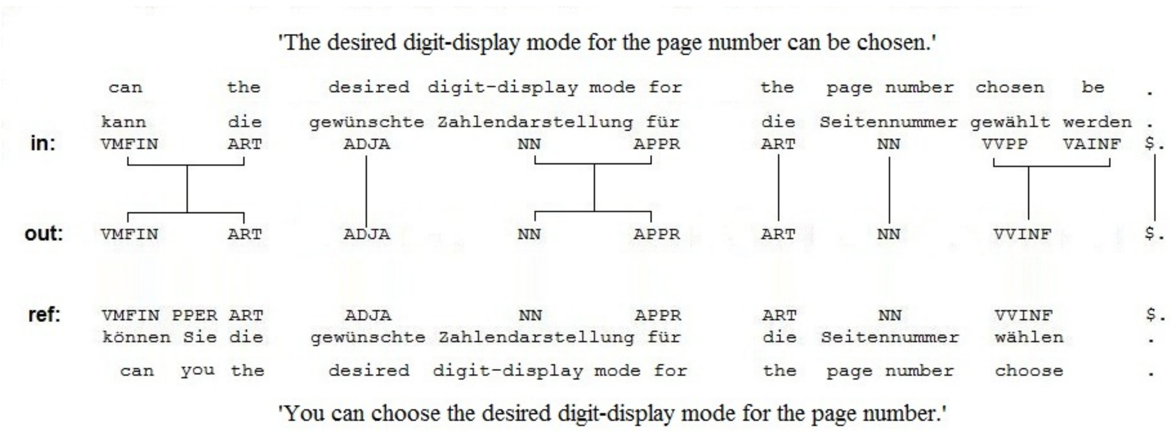
'The desired digit-display mode for the page number can be chosen.'
can the desired digit-display mode for the page number chosen be .
kann die gewünschte Zahlendarstellung für die Seitennummer gewählt werden .
in: VMFIN ART ADJA NN APPR ART NN VVPP VAINF $.
out: VMFIN ART ADJA NN APPR ART NN VVINF $.
ref: VMFIN PPER ART ADJA NN APPR ART NN VVINF $.
können Sie die gewünschte Zahlendarstellung für die Seitennummer wählen .
can you the desired digit-display mode for the page number choose .
'You can choose the desired digit-display mode for the page number.'

Figure 3: Test item #3, with output from the standard phrase-based system, including phrase correspondence

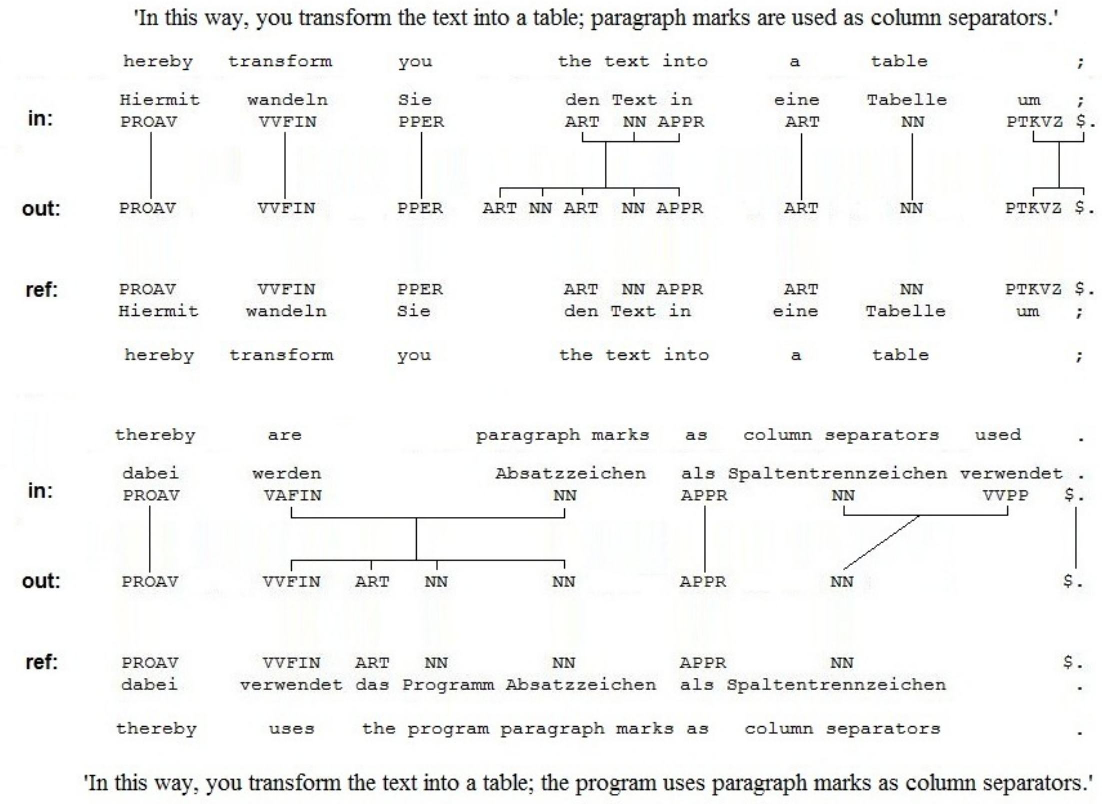
'In this way, you transform the text into a table; paragraph marks are used as column separators.'
hereby transform you the text into a table ;
Hiermit wandeln Sie den Text in eine Tabelle um ;
in: PROAV VVFIN PPER ART NN APPR ART NN PTKVZ $.
out: PROAV VVFIN PPER ART NN ART NN APPR ART NN PTKVZ $.
ref: PROAV VVFIN PPER ART NN APPR ART NN PTKVZ $.
Hiermit wandeln Sie den Text in eine Tabelle um ;
hereby transform you the text into a table ;
thereby are paragraph marks as column separators used .
dabei werden Absatzzeichen als Spaltentrennzeichen verwendet .
in: PROAV VAFIN NN APPR NN VVPP $.
out: PROAV VVFIN ART NN NN APPR NN $.
ref: PROAV VVFIN ART NN NN APPR NN $.
dabei verwendet das Programm Absatzzeichen als Spaltentrennzeichen .
thereby uses the program paragraph marks as column separators .
'In this way, you transform the text into a table; the program uses paragraph marks as column separators.'

Figure 4: Test item #4, split into two lines due to its length, with output from the standard phrase-based system, including phrase correspondence

tical to the reference. However, the deletion of the verb participle `VVPP` (*gestartet* 'started') and the insertion of the semantically related finite verb `VVFIN` (*beginnt* 'begins') occur in different top-level phrases, which leads one to suspect that this success at the POS level would not easily be carried over to a success at the surface level.

4 Our Plan

To avoid the problem of the system performing depassivization steps where there was no passive to begin with, one could try giving the system information on where the *Avoid passives* flag occurred within the sentence. One way would be to treat a token within the flagged region as being of an entirely different class, e.g. a flagged finite auxiliary verb might be `VAFIN_flagged` rather than `VAFIN`.

Another idea for helping the system to learn where not to apply depassivization would be to add training data in which no passive occurs, and thus in which the source and target segments are identical.

The depassivization of German sentences often involves long-distance relationships on the input side which disappear in the output due to the elimination of the auxiliary verb. Braune et al (2012) extend hierarchical SMT with a method to extract an additional and separate set of rules specifically for long-distance reorderings. An SMT depassivizer such as ours may benefit from incorporation of their method. It seems therefore promising to investigate this in future work.

The 517 segment pairs containing depassivization form an excruciatingly small training set by the standards of SMT, so an obvious approach to improving the results is to get more data, which means collecting German passive sentences and depassivizing them by hand.

If an SMT system proves able to depassivize at the POS level, that would give us reason to expect that it could do the same at the surface level given enough data. That said, a POS-level SMT depassivizer could in itself perhaps be useful as a component of an automatic surface-depassivizer that uses heuristics to guess the output words from the alignments between the input words and the output POSes.

5 Conclusions

Using statistical machine translation technology, we produced systems that are sometimes able to depassivize German sentences represented at the part-of-speech level, though not with sufficient consistency to be useful. Nonetheless, our preliminary results show some possibility that this strategy has the potential to be successful. Our results are preliminary since we used a tiny data set consisting of 517, 20, and 10 text segment pairs for training, tuning, and test sets, respectively. We were limited to this size because the data is slow and expensive to produce, as each segment must be edited by a human. We presented ideas for improving the system, and if these prove fruitful and we are able to achieve an automatic German depassivizer, it opens the door to possibly automating the correction of a variety of other grammar and style issues in various languages using the same technique.

References

Fabienne Braune, Anita Gojun, and Alexander Fraser. 2012. Long-distance reordering during search for hierarchical phrase-based SMT. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pp. 177-184, Trento, Italy.

Andrew Bredenkamp, Berthold Crysmann, and Mirela Petrea. 2000. Building Multilingual Controlled Language Performance Checkers. In *Proceedings of the 3rd International Workshop on Controlled Language Applications*, pp. 83-89, Seattle, Washington.

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop On Statistical Machine Translation*, Prague, Czech Republic.

Dmitriy Genzel. 2010. Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp. 376-384, Beijing, China.

Philipp Koehn and Hieu Hoang. June, 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 868-876, Prague, Czech Republic.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi,

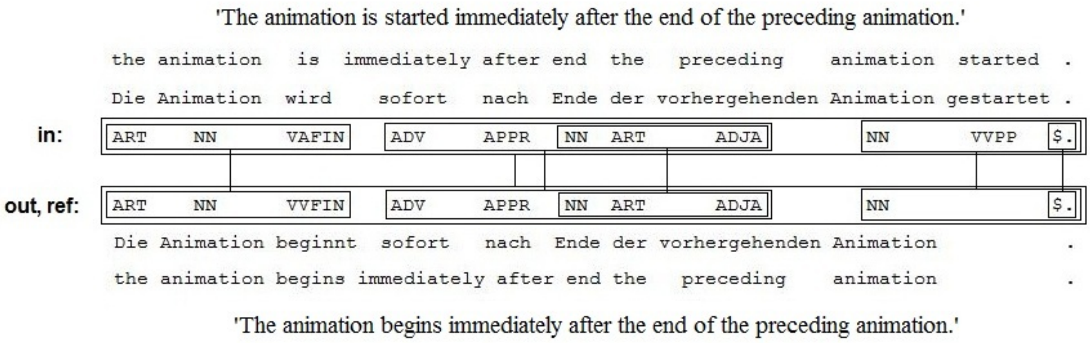
'The animation is started immediately after the end of the preceding animation.'
the animation is immediately after end the preceding animation started .
Die Animation wird sofort nach Ende der vorhergehenden Animation gestartet .
in:
ART NN VAFIN
ADV APPR
NN ART ADJA
NN VVPP
$.
out, ref:
ART NN VVFIN
ADV APPR
NN ART ADJA
NN
$.
Die Animation beginnt sofort nach Ende der vorhergehenden Animation .
the animation begins immediately after end the preceding animation .
'The animation begins immediately after the end of the preceding animation.'

Figure 5: Test item #1, with output from the tree-based system, including phrase correspondence

Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. June, 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics (ACL)*, demonstration session, Prague, Czech Republic.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. July, 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Pennsylvania.

Maja Popović and Hermann Ney. May, 2006. POS-based Word Reorderings for Statistical Machine Translation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pp. 1278-1283, Genoa, Italy.

tekom. 2011. *Regelbasiertes Schreiben: Deutsch für die Technische Kommunikation.*

Jörg Tiedemann. May, 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.