# TQ-AutoTest – A Semi-Automatic Test Suite for (Machine) Translation Quality

**Vivien Macketanz, Renlong Ai, Aljoscha Burchardt and Hans Uszkoreit**
DFKI – Language Technology Lab
Berlin, Germany
{firstname.lastname}@dfki.de

## Abstract

In several areas of NLP evaluation, test suites have been used to analyze the strengths and weaknesses of systems. Today, Machine Translation (MT) quality is usually assessed by shallow automatic comparisons of MT outputs with reference corpora resulting in a number. Especially the trend towards neural MT has renewed peoples' interest in better and more analytical diagnostic methods for MT quality. In this paper we present TQ-AutoTest, a novel framework that supports a linguistic evaluation of (machine) translations using test suites. Our current test suites comprise about 5000 handcrafted test items for the language pair German–English. The framework supports the creation of tests and the semi-automatic evaluation of the MT results using regular expressions. The expressions help to classify the results as correct, incorrect or as requiring a manual check. The approach can easily be extended to other NLP tasks where test suites can be used such as evaluating (one-shot) dialogue systems.

**Keywords:** Machine Translation, Quality Evaluation, Test Suites

## 1. Introduction and Background

In several areas of NLP evaluation, test suites have been used to analyze the strengths and weaknesses of systems. In contrast to "real-life" gold standard corpora, test suites can contain made-up or edited input-output pairs to isolate interesting or difficult phenomena.

In Machine Translation (MT) research, broadly-defined test suites have not been used apart from several singular attempts (King and Falkedal, 1990; Isahara, 1995; Koh et al., 2001, etc.). One of the reasons for this might be the fear that the performance of statistical MT systems depends so much on the particular input data, parameter settings, etc., that relevant conclusions about the errors they make are difficult to obtain. Another concern is that "correct" MT output cannot be specified in the same way as the output of other language processing tasks like parsing or fact extraction where the expected results can be more or less clearly defined. Due to the variation of language, ambiguity, etc., checking and evaluating MT output can be almost as difficult as the translation itself.

Today, MT quality is still usually assessed by shallow automatic comparisons of MT outputs with reference corpora resulting in a number. Early attempts to automatically classify errors based on post-edits or reference translations like (Popović and Ney, 2011) have not yet become standard. For the detection of certain types of errors like grammar errors, parsers have been used (Tezcan et al., 2016). In other narrow domains, researchers have started to explore the differences between systems and between the development stages of one system in more linguistic detail. Especially the trend towards neural MT has renewed peoples' interest in better and more analytical diagnostic methods for MT quality. Recent work based on specific test suites includes the study of verb-particle constructions (Schottmüller and Nivre, 2014), pronouns (Guillou and Hardmeier, 2016) or structural divergences (Isabelle et al., 2017). (Bentivogli et al., 2016) performed a comparison of neural- with phrase-based MT systems on IWSLT data using a coarse-grained error typology where neural systems have been found to make fewer morphological, lexical and word-order errors.

Using our own test suites, we have performed several comparative studies of different MT systems both in the general domain (Burchardt et al., 2017) and in the technical domain (Beyer et al., 2017). When presenting this work, one of the most (obvious) criticism we got was the huge amount of manual effort that was involved in the evaluation procedure. In this paper we will present the novel TQ-AutoTest framework that allows for a drastic reduction of the manual effort when checking translation quality on the basis of test suites.

This article is structured as follows: In Section 2. we will briefly introduce our own test suite and the manual evaluation procedure we have applied in the past. Section 3. describes the new TQ-AutoTest framework that supports the evaluation procedure. A use case of the TQ-AutoTest will be shown in Section 4.. Finally, in Section 5. we will conclude and give an outlook on future work.

## 2. Test Suites for German – English

We have built a test suite for a fine-grained evaluation of MT quality for the language pair German – English. In brief, it contains segments selected from various parallel corpora and drawn from other sources such as grammatical resources, e.g., the TSNLP Grammar Test Suite (Lehmann et al., 1996) and online lists of typical translation errors.

Each test sentence is annotated with a phenomenon category and the phenomenon it represents. An example showing these fields can be seen in Table 1 with the first column containing the source segment and the second and third column containing the phenomenon category and the phenomenon, respectively. The fourth column shows an example machine translation[1] and the last column contains a

[1] As example we have used the "old" Google Translate system that was used before Google changed to a neural system in September 2016, cf. `https://research.googleblog.com/2016/09/a-neural-network-for-machine.html`.

post-edit of the MT output that is created by making as few changes as possible.

In our latest version of the test suite, we have a collection of about 5,000 segments per language direction that are classified in about 15 categories (most of them similar in both language directions) and about 120 phenomena (many of them similar but also some differing, as they are language-specific). Each phenomenon is represented by at least 20 test segments in order to guarantee a balanced test set. The categories cover a wide range of different grammatical aspects that might or might not lead to translation difficulties for a MT system.

### 2.1. Manual Evaluation Procedure

In order to evaluate a system's performance on the categories in the test suite, we concentrate solely on the phenomenon in the respective sentence and disregard other errors. This means that we have to determine whether a translation error can be linked to the phenomenon under examination or if it is independent from the phenomenon. If the former is the case, the segment will be validated as incorrect. If, however, the error in the translation cannot be traced back to the phenomenon, the segment will be counted as correct.

When conducting the manual evaluation, the system outputs were automatically being compared to a "reference translation", which is, in fact, the post-edit of the Google Translate output, as those were the very first translations to be generated and evaluated when we started building the test suite. In a second step, all the translations that did not match the "reference" were manually evaluated by a professional linguist since the translations might be very different from the Google post-edit but nevertheless correct. This is also the reason why we refrained from creating an independent reference. As a consequence, we cannot compute automatic scores like BLEU. However, we do not see this as a disadvantage as with the test suite we want to focus rather on gaining insights about the nature of translations than on how well translations match a certain reference.

Nevertheless, this manual evaluation is a very time-consuming process, especially when dealing with such a large dataset and different MT systems, thus, we decided to come up with a semi-automatic solution, i.e., the TQ-AutoTest.

## 3. The TQ-AutoTest Framework

With the test suite growing bigger over time, we decided to implement a framework that facilitates the evaluation procedure by automating the analysis. Therefore, we built the TQ-AutoTest. In order to include as many correct translations options as possible, the TQ-AutoTest is based on regular expressions (cf. Section 3.1.). Currently, the automation is almost fully completed for the language direction German→English and we are working on expanding and completing the other language direction.

Presently, the TQ-AutoTest exhibits the following features (described in further detail in the following Sections): data preparation; upload report; view report; compare engines; regular expression evaluation; expand, edit and query database (cf. Figures 2 and 3).

With these functions, the TQ-AutoTest can be used for different purposes: You can not only test a system's performance with regard to the linguistic phenomena but also compare the performance of different systems/system types or track changes within one system's performance. By doing so, you can test the system(s) either on all phenomena, or just a selection of the phenomena. To prevent overfitting or cheating, we will not publish the test items. Before sending them to colleagues who want their engines tested, we use a mechanism for scrambling the test segments with a large amount of "distractor" segments.

### 3.1. Regular Expressions

The foundation of the evaluation with the TQ-AutoTest are regular expressions. With the help of these patterns, we try to cover as many correct translations as possible. In line with our manual evaluation procedure briefly described in Section 2.1., the regular expressions only focus on the part of the segment that is under investigation, i.e., the respective phenomenon. Since all other mistranslations that cannot be related to the phenomenon are ignored, it is not necessary for the regular expressions to cover the whole sentence.

The process of creating the regular expressions was thus very complex and elaborate. They have been built manually by a linguist, supported by a professional translator. The regular expressions are based on MT outputs that had been generated before and were then expanded by experience, e.g., which correct/incorrect translation could be expected for a source segment. Considering that once the corpus is completed it can be used over and over again, we are convinced it is worthwhile investing the time and effort to create the regular expressions.

We did not only create positive regular expressions with which the MT output can be evaluated as correct, but in some cases also negative regular expressions with which the MT output is evaluated as incorrect:

Example (1)

| | |
|---|---|
| Source: | Sie fuhr das Auto ihres *Mannes.* |
| Output 1: | She drove her *husband*'s car. |
| Output 2: | She drove the *man*'s car. |
| Output 3: | She drove the *blue* car. |
| positive regex: | husband\|spouse\|hubb(y\|ies) |
| negative regex: | (gentle)?m[ae]n\|guy |

The German source sentence in example (1) contains a lexical ambiguity: The German word *Mann* can either mean *man* or *husband.* In combination with a possessive pronoun (in this case *ihr - her*), *Mann* always refers to *husband.* Output 1 - 3 are examples of different MT outputs. As can be seen, only output 1 matches the positive regular expression. The regular expression also allows translations that include the words *spouse*, *hubby* or *hubbies*[3].

[3] We include the plural of *hubby* as well since the focus in this category lies entirely on the lexical ambiguity of the German word *Mann* and, thus, a translation containing *hubbies* instead of *hubby* or *hubby's* would be evaluated as correct.

| Source | Category | Phenomenon | Example Target (raw) | Target (edited) |
|---|---|---|---|---|
| Lena machte sich früh vom Acker. | MWE | Idiom | Lena [left the field early].[2] | Lena left early. |
| Lisa hat Lasagne gemacht, sie ist schon im Ofen. | Non-verbal agreement | Coreference | Lisa has made lasagne, [she] ist already in the oven. | Lisa has made lasagna, it is already in the oven. |
| Ich habe der Frau das Buch gegeben. | Verb tense/ aspect/mood | Ditransitive - perfect | I [have the woman of the Book]. | I have given the woman the book. |

Table 1: Example test suite entries German → English (simplified for display purposes).

Output 2 on the other hand matches the negative regular expression and thus would be evaluated as incorrect. Output 3 does not match any of the regular expressions and therefore would be reconsidered in a follow-up manual check (cf. Section 3.2.). A screenshot of a positive match with a regular expression in the TQ-AuteTest can be seen in Figure 1.

In order to ensure the syntactical correctness of the regular expressions, the TQ-AutoTest also contains a "RegEx Evaluator" in which regular expressions can be tested for syntactical correctness and completeness. Furthermore, regular expressions can be augmented during the evaluation process.

In addition to the regular expressions, we also implemented a feature for positive and negative tokens. With this feature, a whole sentence (i.e., a MT output) can be added to the database to be matched against in subsequent evaluations. This feature is very convenient for phenomena or segments that are more complex. As a consequence, the database is constantly expanding and covers an increasing amount of possible MT outputs.

Every segment in the database is at least either covered by a regular expression or a negative/positive token, some segments by both. 99% of the segments are covered by a positive regular expression, 40% of the segments are covered by a negative regular expression. Most of the sentences that exhibit a negative regular expression do also feature a positive regular expression. Furthermore, 5% of the segments are currently covered by one or more positive tokens while 48% of the segments are covered by negative tokens. Especially the number of segments featuring negative tokens are increasing with every report when new erroneous translations are evaluated. Thus, the database is constantly growing and the more reports are carried out, the less manual work of inspecting segments that are neither covered by a regular expression nor by a positive/negative token is needed.

### 3.2. Workflow

A typical workflow in the TQ-AutoTest looks as follows (not all steps must necessarily be realized):

**Data Preparation** An absolute or relative number of sentences from all categories/a selection of categories, resp. from all phenomena/a selection of phenomena is selected randomly (cf. Figure 2) and then scrambled with a random selection of the distractors, whereby the scramble factor can be selected manually. The resulting data is generated in a text file with an ID.

This text file can then be used for running the translations, e.g., on different types of MT systems, say a phrase-based and a neural MT system, or on different version of one system, e.g., before and after some expected improvement.

**Upload Report** Once the translations are generated (the order of the sentences must be maintained), a text file with the outputs can be uploaded. With the upload, information about the engine (e.g., Google), the type of engine (e.g., NMT) and further comments must/can be entered, cf. Figure 3.

The test sentences are then automatically unscrambled from the distractors and the sentences are evaluated based on the database of regular expressions and tokens.

**View Reports** In this tab, all reports that have been generated can be viewed and edited. Edited means in this case that sentences that did not match any of the regular expressions or tokens need to be double-checked manually. Correct outputs are shaded in green, incorrect outputs in red and outputs that need to be determined are shaded in yellow. If desired, the evaluation of the manual checking can be added to the database ("Apply Tokens", if it should not be added "Skip Tokens"), cf. Figure 4.

Furthermore, a statistic about the amount of correct/incorrect/tbd translations is automatically generated. This statistic contains tables as well as graphs, both of which can be exported.

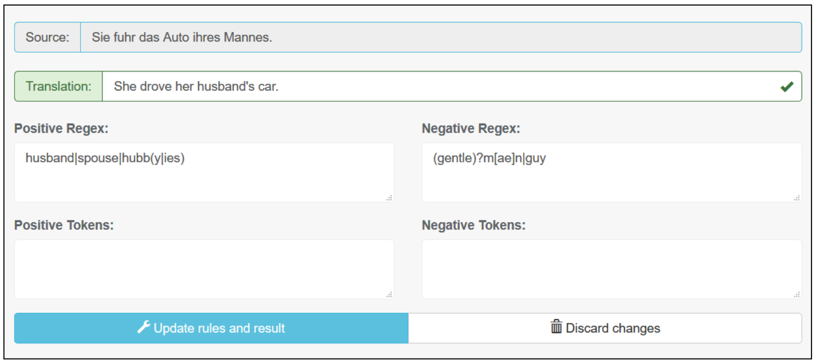
**Compare Engines** This function allows for a comparison of different MT systems/system types that generated translations for the same (sub)set of sentences. Hereby, the absolute and relative numbers of correct/incorrect/tbd translations per systems are calculated (1) on the phenomena, (2) on the categories and (3) on average, and are displayed in tables as well as graphs, both of which can be exported as well.

An exported graph with five different MT systems can be found in Section 4. as an example.

### 3.3. Implementation

The web interface is implemented using Play Framework, which is open-source, reactive, flexible, and provides Typesafe so that both research and commercial requirements are supported. The front-end uses bootstrap library to ensure compatibility across browsers and platforms, and the back-end is implemented with

[3] Square brackets have been added manually to show the erroneous parts.



Source: Sie fuhr das Auto ihres Mannes.

Translation: She drove her husband's car. ✓

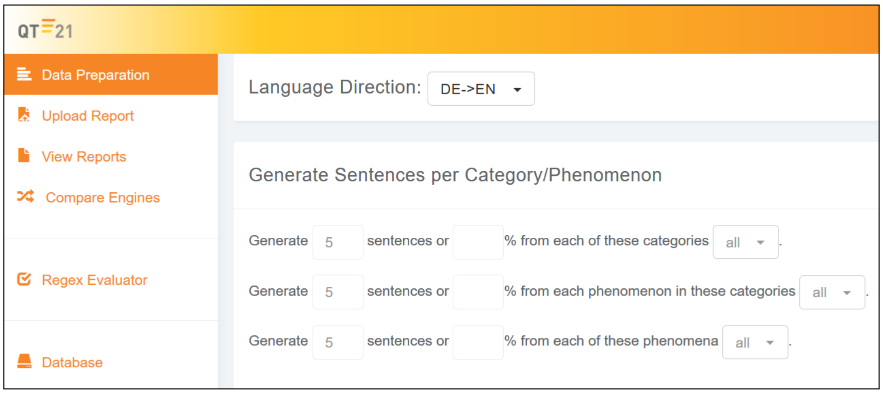
Positive Regex: husband|spouse|hubb(y|ies)

Negative Regex: (gentle)?m[ae]n|guy

Positive Tokens:

Negative Tokens:

Figure 1: Screenshot: Positive Match with RegEx.



Language Direction: DE->EN

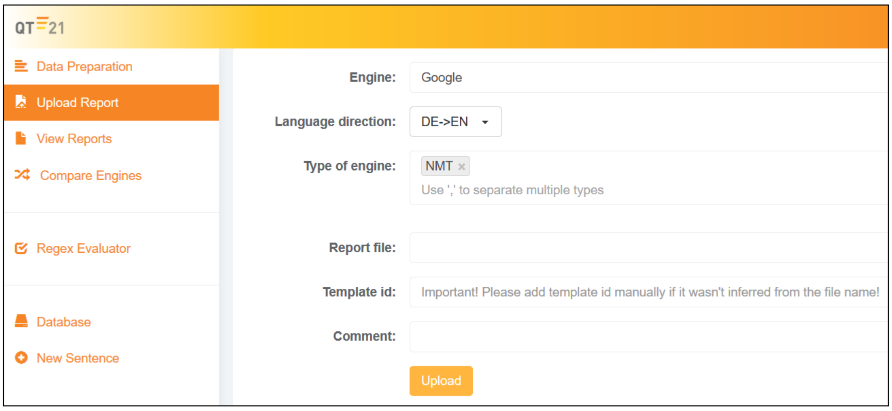
Generate Sentences per Category/Phenomenon

Generate 5 sentences or % from each of these categories all .

Generate 5 sentences or % from each phenomenon in these categories all .

Generate 5 sentences or % from each of these phenomena all .

Figure 2: Screenshot: Data Preparation.



Engine: Google

Language direction: DE->EN

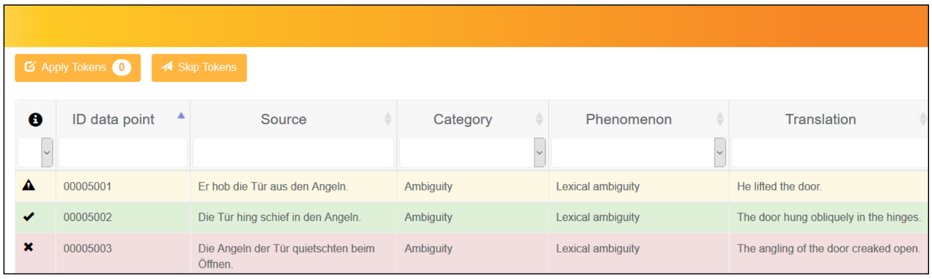
Type of engine: NMT ×
Use ',' to separate multiple types

Report file:

Template id: Important! Please add template id manually if it wasn't inferred from the file name!

Comment:

Figure 3: Screenshot: Upload Report.



| ℹ | ID data point | Source | Category | Phenomenon | Translation |
|---|---|---|---|---|---|
| ▲ | 00005001 | Er hob die Tür aus den Angeln. | Ambiguity | Lexical ambiguity | He lifted the door. |
| ✓ | 00005002 | Die Tür hing schief in den Angeln. | Ambiguity | Lexical ambiguity | The door hung obliquely in the hinges. |
| ✗ | 00005003 | Die Angeln der Tür quietschten beim Öffnen. | Ambiguity | Lexical ambiguity | The angling of the door creaked open. |

Figure 4: Screenshot: View Reports.

Scala. Both templates and test results are stored in Mysql database. The software can be downloaded at `https://gitlab.com/QT21/QT21-Resources/tree/master/Tools/TQ-AutoTest`. Note that the test items themselves are not part of the available resources as the purpose of the test set is to test different systems on the same test set.

## 4. Example of Use

To exemplify TQ-AutoTest, we have compared of the performance of five different MT systems "as-is". The systems we investigated are (1) a neural MT system built by the University of Edinburgh[4], (2) the "old" Google Translate phrase-based statistical MT system (cf. Section 2.), (3) the "new" Google Translate NMT system, (4) the DeepL NMT system[5] and (5) the rule-based MT system Lucy in a completely unadapted version (Alonso and Thurmair, 2003).

The results of the comparison can be found in Figure 5 and 6. Percentage values in boldface indicate that the respective system is significantly better on the particular phenomenon under investigation with a 0.95 confidence level. We calculated the statistical significances by means of a Z-test.

The numbers of instances of segments on the different categories vary strongly. This is due to the fact that we wanted to include the PBMT Google Translate system and we did not have translations of all the segments that are now included in our TQ-AutoTest from this system. We are aware of the fact that the high variety in the numbers of instances (especially the high number of instances in the category verb tense/aspect/mood) creates a bias in the average score. For example, the unadapted Lucy system achieves the highest average score. This high average score is linked to the fact that Lucy is the best-performing system on the category verb tense/aspect/mood. Since Lucy is a rule-based system, many of the rules regarding verb paradigms are probably implemented in the grammar of the system.

Again, we want to stress that it is not our goal to find the "once-and-for-all winning system" with this comparison. The numbers shown here do not represent the corpus frequency of the phenomena. They do solely show tendencies the systems reveal towards the categories that we have tested. Our goal is to provide analytical insights into the systems' strengths and weaknesses in terms of (linguistic) phenomena.

The DeepL system, which is a quite new system, is being promoted as generating better MT outputs than the current Google Translate system. Our numbers support this claim. The Edinburgh NMT system is an almost equal competitor, coming close to the Google NMT average. The Google Translate PBMT system, however, cannot compete with the other systems in this experiment and is the only system that has an average score of less than 50%.

Turning to the scores of the different categories, it becomes clear that the systems perform quite differently on the grammatical phenomena. There are categories in which all of the systems perform quite similar, as for example named entity & terminology. In this category, the scores range from 70.2% to 78.6 %. In other categories as long distance dependency (LDD) & interrogatives on the other hand, the scores range from 39.0% to 77.3%.

These insights can now serve as inspiration for developers to modify the systems (or the training data) in order to improve their performance. We take the findings introduced here as evidence that it is time to complement the reference-based evaluation of MT systems that work well in the laboratory with a reference-independent, more analytical evaluation that can be applied in situations where one does not have full control over the systems, test corpora, where no references are available or where one wants to compare systems just as they are. With our TQ-AutoTest, we provide a tool that semi-automates this more complex evaluation procedure.

## 5. Conclusion and Outlook

In this paper we have presented TQ-AutoTest, a framework that supports the analytical evaluation of (machine) translations using test suites. Our current test suites comprise about 5000 test items for the language pair German–English in both directions. The framework supports the creation of tests and the evaluation of the translation results using regular expressions. The expressions classify the results as correct, incorrect or requiring a manual check. From our experience, all errors can be checked with regular expressions. As one would expect, e.g, word errors are easier to code than, e.g., grammatical errors. For verb paradigms, we list all possible sentences in the regular expressions as this has turned out to be easier than creating complex regular expressions. We see this as the beginning of more research in the direction we have indicated.

Seen that in previous experiments, we have classified all errors fully manually, the regular expressions provide a drastic reduction of manual labor. After finalizing the regular expressions, we will conduct more tests of the tool. Looking into the future, the approach allows for a number of extensions. Obvious possibilities are more languages and including also domain-specific test suites. Both will be manual work, but given the experience and example we have created will hopefully speed up the process.

It is also imaginable to extend the approach to other NLP applications such as dialogue (Chatbots). We have a concrete request by an industry partner to explore the possibility of evaluating meeting translations that we are currently pursuing.

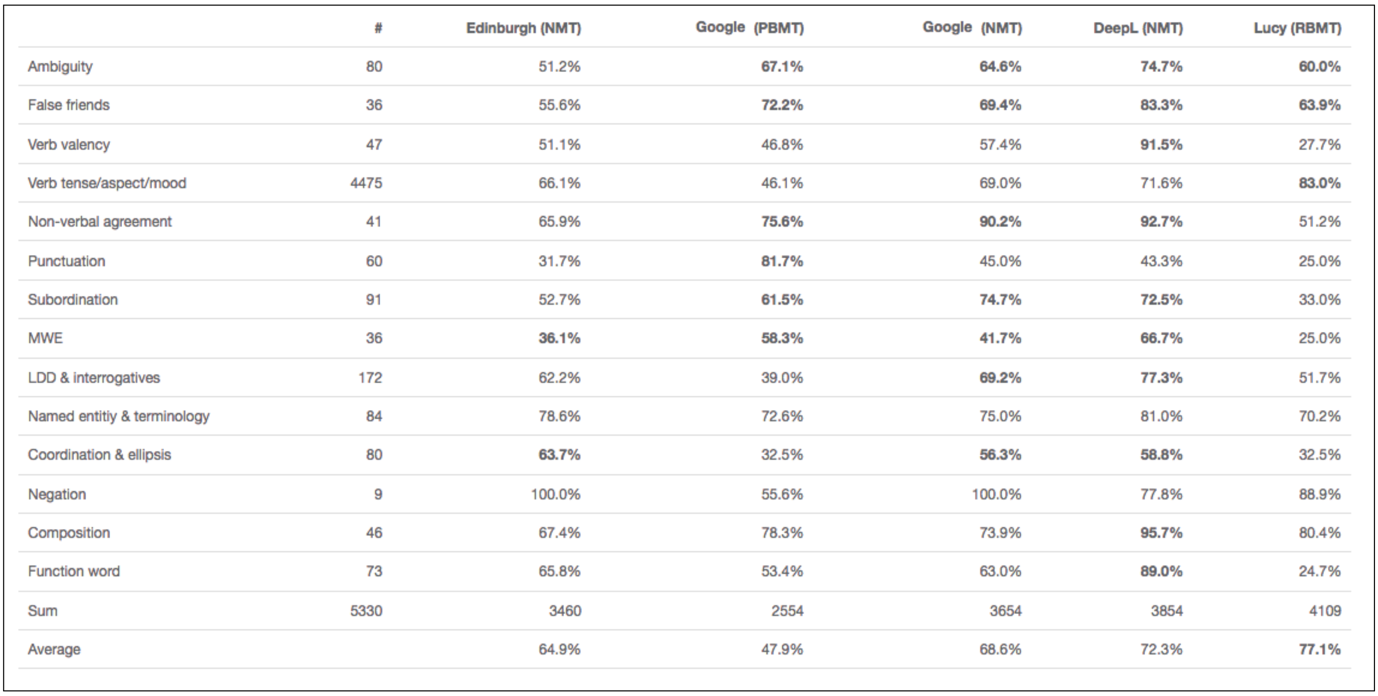
## 6. Acknowledgements

We heartily thank Prof. Silvia Hansen-Schirra (Johannes Gutenberg-Universität Mainz) and her team as well as Kim Harris (text & form) for their very valuable support in the creation of the test suite. We also thank our colleague He Wang for helping with the implementation of the tool.

This article has received support from the EC's Horizon 2020 research and innovation programme under grant agreements no. 645452 (QT21) and by the German Federal Ministry of Education and Research (BMBF) through the project BBDC (01IS14013E).

[4] A detailed description of the setup of the system can be found in (Sennrich et al., 2016).

[5] `https://www.deepl.com/translate`



| | # | Edinburgh (NMT) | Google (PBMT) | Google (NMT) | DeepL (NMT) | Lucy (RBMT) |
|---|---|---|---|---|---|---|
| Ambiguity | 80 | 51.2% | **67.1%** | **64.6%** | **74.7%** | **60.0%** |
| False friends | 36 | 55.6% | **72.2%** | **69.4%** | **83.3%** | **63.9%** |
| Verb valency | 47 | 51.1% | 46.8% | 57.4% | **91.5%** | 27.7% |
| Verb tense/aspect/mood | 4475 | 66.1% | 46.1% | 69.0% | 71.6% | **83.0%** |
| Non-verbal agreement | 41 | 65.9% | **75.6%** | **90.2%** | **92.7%** | 51.2% |
| Punctuation | 60 | 31.7% | **81.7%** | 45.0% | 43.3% | 25.0% |
| Subordination | 91 | 52.7% | **61.5%** | **74.7%** | **72.5%** | 33.0% |
| MWE | 36 | **36.1%** | **58.3%** | **41.7%** | **66.7%** | 25.0% |
| LDD & interrogatives | 172 | 62.2% | 39.0% | **69.2%** | **77.3%** | 51.7% |
| Named entitiy & terminology | 84 | 78.6% | 72.6% | 75.0% | 81.0% | 70.2% |
| Coordination & ellipsis | 80 | **63.7%** | 32.5% | **56.3%** | **58.8%** | 32.5% |
| Negation | 9 | 100.0% | 55.6% | 100.0% | 77.8% | 88.9% |
| Composition | 46 | 67.4% | 78.3% | 73.9% | **95.7%** | 80.4% |
| Function word | 73 | 65.8% | 53.4% | 63.0% | **89.0%** | 24.7% |
| Sum | 5330 | 3460 | 2554 | 3654 | 3854 | 4109 |
| Average | | 64.9% | 47.9% | 68.6% | 72.3% | **77.1%** |

Figure 5: Exported table: compared Engines on the categories.

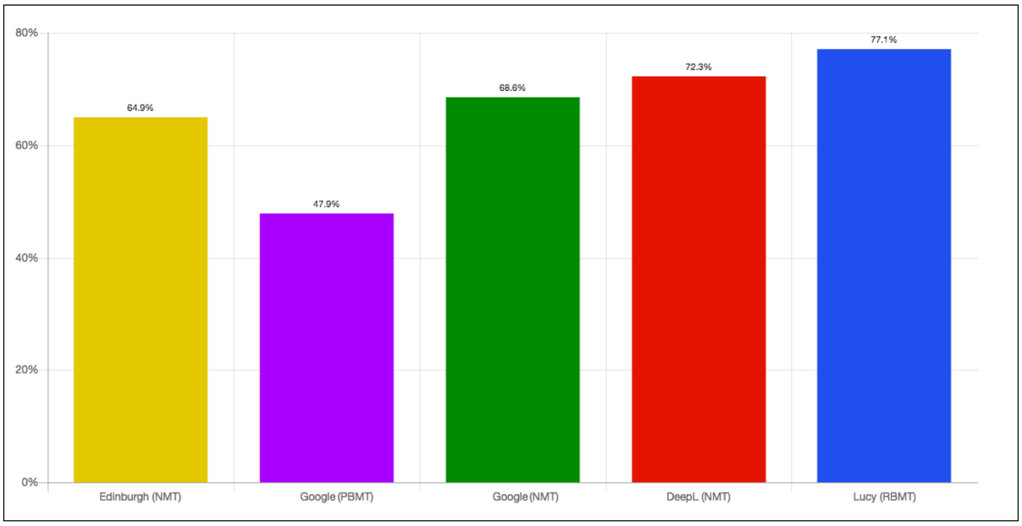


Figure 6: Exported graph: average values of compared engines.

## 7. Bibliographical References

Alonso, J. A. and Thurmair, G. (2003). The Comprendium Translator system. In *Proceedings of the Ninth Machine Translation Summit*, New Orleans, Louisiana, USA, September.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2016). Neural versus phrase-based machine translation quality: a case study. *CoRR*, abs/1608.04631.

Beyer, A., Macketanz, V., Williams, P., and Burchardt, A. (2017). Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data? In *Proceedings of the 20th Annual Conference of the European Association for Machine Translation (EAMT): User Studies and Project/Product Descriptions*, pages 41–46, Prague, Czech Republic, May.

Burchardt, A., Macketanz, V., Dehdari, J., Heigold, G., Peter, J.-T., and Williams, P. (2017). A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108(1):159–170.

Guillou, L. and Hardmeier, C. (2016). Protest: A test suite for evaluating pronouns in machine translation. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Isabelle, P., Cherry, C., and Foster, G. F. (2017). A Challenge Set Approach to Evaluating Machine Translation. *CoRR*, abs/1704.07431.

Isahara, H. (1995). Jeida's test-sets for quality evaluation of mt systems: Technical evaluation from the developer's point of view. In *Proceedings of the MT Summit V. Luxembourg*.

King, M. and Falkedal, K. (1990). Using test suites in evaluation of machine translation systems. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 2*, COLING '90, pages 211–216, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koh, S., Maeng, J., Lee, J.-Y., Chae, Y.-S., and Choi, K.-S. (2001). A test suite for evaluation of english-to-korean machine translation systems. In *Proceedings of the MT Summit VIII. Santiago de Compostela, Spain*.

Lehmann, S., Oepen, S., Regnier-Prost, S., Netter, K., Lux, V., Klein, J., Falkedal, K., Fouvry, F., Estival, D., Dauphin, E., Compagnion, H., Baur, J., Balkan, L., and Arnold, D. (1996). TSNLP - Test Suites for Natural Language Processing. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 711–716, Copenhagen, Denmark, August.

Popović, M. and Ney, H. (2011). Towards automatic error analysis of machine translation output. *Computational Linguistics*, 37(4):657–688, December.

Schottmüller, N. and Nivre, J. (2014). Issues in translating verb-particle constructions from german to english. In *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*, pages 124–131, Gothenburg, Sweden, April. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT)*, pages 371–376, Berlin, Germany, August.

Tezcan, A., Hoste, V., and Macken, L. (2016). Detecting grammatical errors in machine translation output using dependency parsing and treebank querying. *BALTIC JOURNAL OF MODERN COMPUTING*, 4(2):203–217.