The Prague Bulletin of Mathematical Linguistics
NUMBER 98 OCTOBER 2012 25–35

Appraise: an Open-Source Toolkit for Manual Evaluation of MT Output

Christian Federmann

DFKI Language Technology Lab

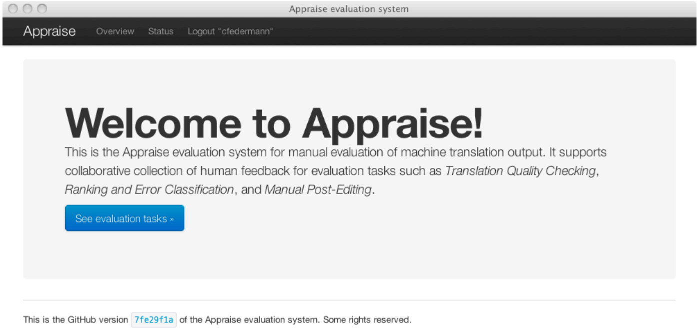
Abstract

We describe Appraise, an open-source toolkit supporting manual evaluation of machine translation output. The system allows to collect human judgments on translation output, implementing annotation tasks such as 1) quality checking, 2) translation ranking, 3) error classification, and 4) manual post-editing. It features an extensible, XML-based format for import/ export and can easily be adapted to new annotation tasks. The current version of Appraise also includes automatic computation of inter-annotator agreements allowing quick access to evaluation results. Appraise is actively developed and used in several MT projects.

1. Introduction

Evaluation of Machine Translation (MT) output to assess translation quality is a difficult task. There exist automatic metrics such as BLEU (Papineni et al., 2002) or Meteor (Denkowski and Lavie, 2011) which are widely used in minimum error rate training (Och, 2003) for tuning of MT systems and as evaluation metric for shared tasks such as, e.g., the Workshop on Statistical Machine Translation (WMT) (Callison-Burch et al., 2012). The main problem in designing automatic quality metrics for MT is to achieve a high correlation with human judgments on the same translation output. While current metrics show promising performance in this respect, manual inspection and evaluation of MT results is still equally important as it allows for a more targeted and detailed analysis of the given translation output. The manual analysis of a given, machine translated text is a time-consuming and laborious process; it involves training of annotators, requires detailed and clear-cut annotation guidelines,

Corresponding author: cfedermann@dfki.de
Cite as: Christian Federmann. Appraise: an Open-Source Toolkit for Manual Evaluation of MT Output. The Prague Bulletin of Mathematical Linguistics No. 98, 2012, pp. 25–35. doi: 10.2478/v10108-012-0006-9.



Welcome to Appraise!

This is the Appraise evaluation system for manual evaluation of machine translation output. It supports collaborative collection of human feedback for evaluation tasks such as *Translation Quality Checking*, *Ranking and Error Classification*, and *Manual Post-Editing*.

This is the GitHub version `7fe29f1a` of the Appraise evaluation system. Some rights reserved.

Figure 1. Front page

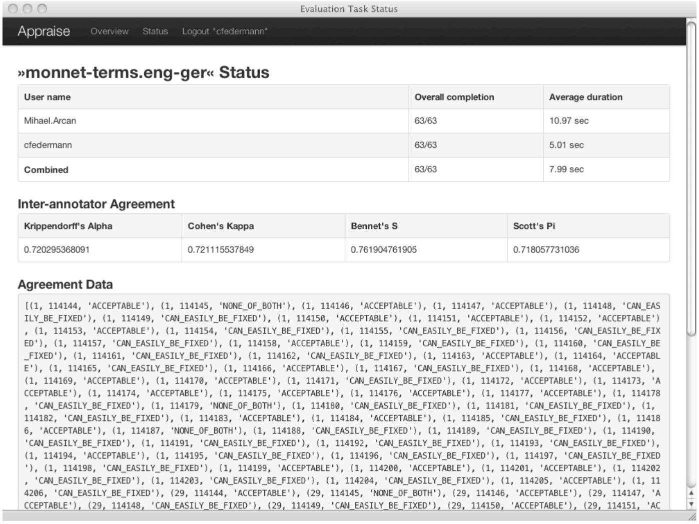
and—last but not least—an annotation software that allows annotators to get their job done quickly and efficiently.

In this paper, we describe *Appraise*, an open-source tool that allows to perform manual evaluation of Machine Translation output. Appraise can be used to collect human judgments on translation output, implementing several annotation tasks. We will describe the tool in more detail on the following pages. The remainder of this paper is structured as follows: Section 2 gives some further motivation concerning the development of the tool before we describe the system in more detail in Section 3 and highlight the various annotation tasks we implemented in Section 4. We explain the installation requirements in Section 5 and give some quick usage instructions in Section 6. Finally, we describe several experiments where Appraise has proven useful (see Section 7) and give some concluding remarks in Section 8.

2. Motivation

As we have mentioned before, the collection of manual judgments on machine translation output is a tedious task; this holds for simple tasks such as translation ranking but also for more complex challenges like word-level error analysis or post-editing of translation output. Annotators tend to lose focus after several sentences, resulting in reduced intra-annotator agreement and increased annotation time. In our experience with manual evaluation campaigns it has shown that a well-designed annotation tool can help to overcome these issues.

Development of the Appraise software package started back in 2009 as part of the EuroMatrixPlus project where the tool was used to quickly compare different sets of candidate translations from our hybrid machine translation engine to get an indication whether our system improved or degraded in terms of translation quality. A first version of Appraise was released and described by Federmann (2010).



»monnet-terms.eng-ger« Status

User name	Overall completion	Average duration
Mihael.Arcan	63/63	10.97 sec
cfedermann	63/63	5.01 sec
Combined	63/63	7.99 sec

Inter-annotator Agreement

Krippendorff's Alpha	Cohen's Kappa	Bennet's S	Scott's Pi
0.720295368091	0.721115537849	0.761904761905	0.718057731036

Agreement Data

```
[(1, 114144, 'ACCEPTABLE'), (1, 114145, 'NONE_OF_BOTH'), (1, 114146, 'ACCEPTABLE'), (1, 114147, 'ACCEPTABLE'), (1, 114148, 'CAN_EAS
ILY_BE_FIXED'), (1, 114149, 'CAN_EASILY_BE_FIXED'), (1, 114150, 'ACCEPTABLE'), (1, 114151, 'ACCEPTABLE'), (1, 114152, 'ACCEPTABLE')
, (1, 114153, 'ACCEPTABLE'), (1, 114154, 'CAN_EASILY_BE_FIXED'), (1, 114155, 'CAN_EASILY_BE_FIXED'), (1, 114156, 'CAN_EASILY_BE_FIX
ED'), (1, 114157, 'CAN_EASILY_BE_FIXED'), (1, 114158, 'ACCEPTABLE'), (1, 114159, 'CAN_EASILY_BE_FIXED'), (1, 114160, 'CAN_EASILY_BE
_FIXED'), (1, 114161, 'CAN_EASILY_BE_FIXED'), (1, 114162, 'CAN_EASILY_BE_FIXED'), (1, 114163, 'ACCEPTABLE'), (1, 114164, 'ACCEPTABL
E'), (1, 114165, 'CAN_EASILY_BE_FIXED'), (1, 114166, 'ACCEPTABLE'), (1, 114167, 'CAN_EASILY_BE_FIXED'), (1, 114168, 'ACCEPTABLE'),
(1, 114169, 'ACCEPTABLE'), (1, 114170, 'ACCEPTABLE'), (1, 114171, 'CAN_EASILY_BE_FIXED'), (1, 114172, 'ACCEPTABLE'), (1, 114173, 'A
CCEPTABLE'), (1, 114174, 'ACCEPTABLE'), (1, 114175, 'ACCEPTABLE'), (1, 114176, 'ACCEPTABLE'), (1, 114177, 'ACCEPTABLE'), (1, 114178
, 'CAN_EASILY_BE_FIXED'), (1, 114179, 'NONE_OF_BOTH'), (1, 114180, 'CAN_EASILY_BE_FIXED'), (1, 114181, 'CAN_EASILY_BE_FIXED'), (1,
114182, 'CAN_EASILY_BE_FIXED'), (1, 114183, 'ACCEPTABLE'), (1, 114184, 'ACCEPTABLE'), (1, 114185, 'CAN_EASILY_BE_FIXED'), (1, 11418
6, 'ACCEPTABLE'), (1, 114187, 'NONE_OF_BOTH'), (1, 114188, 'CAN_EASILY_BE_FIXED'), (1, 114189, 'CAN_EASILY_BE_FIXED'), (1, 114190,
'CAN_EASILY_BE_FIXED'), (1, 114191, 'CAN_EASILY_BE_FIXED'), (1, 114192, 'CAN_EASILY_BE_FIXED'), (1, 114193, 'CAN_EASILY_BE_FIXED'),
(1, 114194, 'ACCEPTABLE'), (1, 114195, 'CAN_EASILY_BE_FIXED'), (1, 114196, 'CAN_EASILY_BE_FIXED'), (1, 114197, 'CAN_EASILY_BE_FIXED
'), (1, 114198, 'CAN_EASILY_BE_FIXED'), (1, 114199, 'ACCEPTABLE'), (1, 114200, 'ACCEPTABLE'), (1, 114201, 'ACCEPTABLE'), (1, 114202
, 'CAN_EASILY_BE_FIXED'), (1, 114203, 'CAN_EASILY_BE_FIXED'), (1, 114204, 'CAN_EASILY_BE_FIXED'), (1, 114205, 'ACCEPTABLE'), (1, 11
4206, 'CAN_EASILY_BE_FIXED'), (29, 114144, 'ACCEPTABLE'), (29, 114145, 'NONE_OF_BOTH'), (29, 114146, 'ACCEPTABLE'), (29, 114147, 'A
CCEPTABLE'), (29, 114148, 'CAN_EASILY_BE_FIXED'), (29, 114149, 'CAN_EASILY_BE_FIXED'), (29, 114150, 'ACCEPTABLE'), (29, 114151, 'AC
```

Figure 2. Individual task status

3. System Description

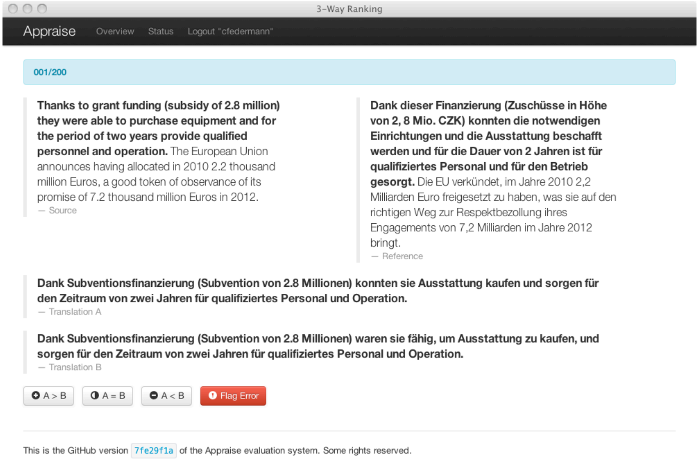
In a nutshell, Appraise is an open-source tool for manual evaluation of machine translation output. It allows to collect human judgments on given translation output, implementing annotation tasks such as (but not limited to):

- translation quality checking;
- ranking of translations;
- error classification;
- manual post-editing.

We will provide a more detailed discussion of these tasks in Section 4.

The software features an extensible XML import/output format and can easily be adapted to new annotation tasks. An example of this XML format is depicted in Figure 5. The software also includes automatic computation of inter-annotator agreement scores, allowing quick access to evaluation results. A screenshot of the task status view is shown in Figure 2. We currently support computation of the following inter-annotator agreement scores:

- Krippendorff's α as described by Krippendorff (2004);
- Fleiss' κ as published in Fleiss (1971), extending work from Cohen (1960);
- Bennett, Alpert, and Goldstein's S as defined in Bennett et al. (1954);
- Scott's π as introduced in Scott (1955).



Thanks to grant funding (subsidy of 2.8 million) they were able to purchase equipment and for the period of two years provide qualified personnel and operation. The European Union announces having allocated in 2010 2.2 thousand million Euros, a good token of observance of its promise of 7.2 thousand million Euros in 2012.
— Source

Dank dieser Finanzierung (Zuschüsse in Höhe von 2, 8 Mio. CZK) konnten die notwendigen Einrichtungen und die Ausstattung beschafft werden und für die Dauer von 2 Jahren ist für qualifiziertes Personal und für den Betrieb gesorgt. Die EU verkündet, im Jahre 2010 2,2 Milliarden Euro freigesetzt zu haben, was sie auf den richtigen Weg zur Respektbezollung ihres Engagements von 7,2 Milliarden im Jahre 2012 bringt.
— Reference

Dank Subventionsfinanzierung (Subvention von 2.8 Millionen) konnten sie Ausstattung kaufen und sorgen für den Zeitraum von zwei Jahren für qualifiziertes Personal und Operation.
— Translation A

Dank Subventionsfinanzierung (Subvention von 2.8 Millionen) waren sie fähig, um Ausstattung zu kaufen, und sorgen für den Zeitraum von zwei Jahren für qualifiziertes Personal und Operation.
— Translation B

This is the GitHub version `7fe29f1a` of the Appraise evaluation system. Some rights reserved.

Figure 3. 3-way ranking task

Agreement computation relies on code from the NLTK project (Bird et al., 2009). Additional agreement metrics can be added easily; the visualisation of agreement scores or other annotation results can be adapted to best match the corresponding annotation task design.

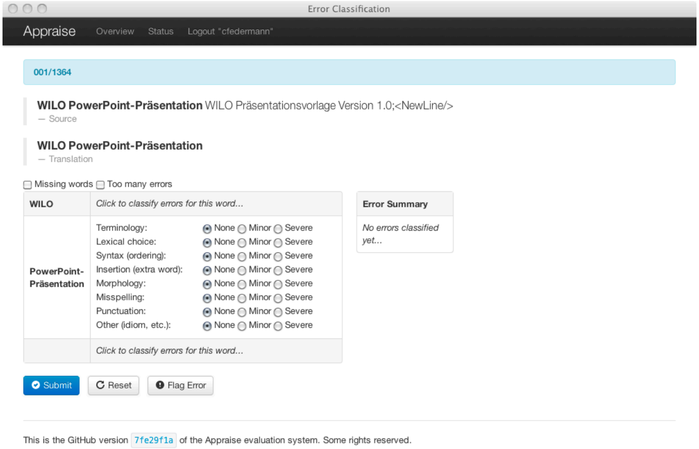
Appraise has been implemented using the Python-based *Django web framework*[1] which takes care of low-level tasks such as "HTTP handling", database modeling, and object-relational mapping. Figures 1–4 show several screenshots of the Appraise interface. We used Twitter's *Bootstrap*[2] as basis for the design of the application and implemented it using long-standing and well-established open-source software with large communities supporting them in the hope that this will also benefit the Appraise software package in the long run.

In the same spirit, we have opened up Appraise development and released the source code on GitHub at `https://github.com/cfedermann/Appraise`. Anybody with a free GitHub account may fork the project and create an own version of the software. Due to the flexibility of the `git` source code management system, it is easy to re-integrate external changes into the master repository, allowing other developers to feed back bug fixes and new features, thus improving and extending the original software. Appraise is available under an open, BSD-style license.[3]

[1] See `http://www.djangoproject.com/` for more information

[2] Available from `http://twitter.github.com/bootstrap/`

[3] See `https://raw.github.com/cfedermann/Appraise/master/appraise/LICENSE`



001/1364

WILO PowerPoint-Präsentation WILO Präsentationsvorlage Version 1.0;<NewLine/>
— Source

WILO PowerPoint-Präsentation
— Translation

□ Missing words □ Too many errors

WILO	*Click to classify errors for this word...*
PowerPoint-Präsentation	Terminology: ● None ○ Minor ○ Severe Lexical choice: ● None ○ Minor ○ Severe Syntax (ordering): ● None ○ Minor ○ Severe Insertion (extra word): ● None ○ Minor ○ Severe Morphology: ● None ○ Minor ○ Severe Misspelling: ● None ○ Minor ○ Severe Punctuation: ● None ○ Minor ○ Severe Other (idiom, etc.): ● None ○ Minor ○ Severe
	Click to classify errors for this word...

Error Summary

No errors classified yet...

This is the GitHub version `7fe29f1a` of the Appraise evaluation system. Some rights reserved.

Figure 4. Error classification task

4. Annotation Tasks

We have developed several annotation tasks which are useful for MT evaluation. All of these have been tested and used during the experiments described in Section 7. The following task types are available for the GitHub version of Appraise:

1. **Ranking** The annotator is shown 1) the source sentence and 2) several ($n \geq 2$) candidate translations. It is also possible to additionally present the reference translation. Wherever available, one sentence of left/right context is displayed to support the annotator during the ranking process.

   We also have implemented a special *3-way ranking task* which works for pairs of candidate translations and gives the annotator an intuitive interface for quick $A > B$, $A = B$, or $A < B$ classification. Figure 3 shows a screenshot of the 3-way ranking interface.

2. **Error Classification** The annotator sees 1) the source (or target) sentence and 2) a candidate translation which has to be inspected wrt. errors contained in the translation output. We use a refined version of the classification described in (Vilar et al., 2006). Error annotation is possible on the sentence level as well as for individual words. The annotator can choose to skip translations containing "too many errors" and is able to differentiate between "minor" and "severe" errors. Figure 4 shows a screenshot of the error classification interface.

3. **Quality Estimation** The annotator is given 1) the source sentence and 2) one candidate translation which has to be classified as *Acceptable*, *Can easily be fixed*, or *None of both*. We also show the reference sentence and again present left/right context if available. This task can be used to get a quick estimate on the *acceptability* of a set of translations.

4. **Post-editing** The annotator is shown 1) the source sentence including left/right context wherever available and 2) one or several candidate translation. The task is defined as choosing the translation which is "easiest to post-edit" and then performing the post-editing operation on the selected translation.

   In some of our experiments with Appraise, we found that annotators did not necessarily choose the overall best candidate translation for post-editing but often selected worse translations which, however, could be post-edited more quickly. Our findings are summarised in Avramidis et al. (2012).

5. Installation Requirements

Appraise requires Python 2.7.x and Django 1.4.x to be installed on the deployment machine. You can install Python using the following commands:

```
$ wget http://www.python.org/ftp/python/2.7.3/Python-2.7.3.tgz
$ tar xzf Python-2.7.3.tgz
$ cd Python-2.7.3
$ ./configure && make && make install
```

After having set up Python, you have to download, extract, and install the Django web framework. This will be installed into the `site-packages` folder that belongs to the `python` binary used to start `setup.py`. Run the following commands:

```
$ wget djangoproject.com/download/1.4/tarball/ -O Django-1.4.1.tar.gz
$ tar xzvf Django-1.4.1.tar.gz
$ cd Django-1.4.1
$ python2.7 setup.py install
```

Note: on Mac OS X, you can also use MacPorts[4] to install Python and Django, simplifying the whole installation procedure to a single command:

```
$ sudo port install py27-django
```

Finally, you have to create a local copy of the Appraise source code package which is available from GitHub. In `git` terminology, you have to "clone" Appraise. You can do so as follows (change `Appraise-Software` to any other folder name you like):

[4] Available from `http://www.macports.org/`

```
$ git clone git://github.com/cfedermann/Appraise.git Appraise—Software
Cloning into 'Appraise—Software'...
...
$ cd Appraise—Software
```

Congratulations! You have just installed Appraise on your local machine.

6. Usage Instructions

Assuming you have already installed Python and Django, and have cloned a local copy of Appraise, you can setup the SQLite database and subsequently start up the server using the following commands:

```
$ cd Appraise—Software/appraise
$ python manage.py syncdb
...
```

When asked whether you want to create a super user account, reply `yes` and create such an account; this will be the administrative user having all permissions.

```
$ python manage.py runserver
Validating models...

0 errors found
Django version 1.4.1, using settings 'appraise.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL—C.
```

You should be greeted with the output shown above in your terminal. In case of any errors during startup, these will be reported instead and, depending on the severity of the problem, Django will refuse to launch Appraise. Point your browser to `http://127.0.0.1:8000/appraise/` and check if you can see the Appraise front page, which looks similar to the screenshot depicted in Figure 1.

New user accounts can be created inside Django's administration backend. You have to login and access `http://127.0.0.1:8000/appraise/admin/auth/user/add/` for user administration. Evaluation tasks are created in the administration backend at `http://127.0.0.1:8000/appraise/admin/evaluation/evaluationtask/add/`. You need an XML file in proper format to upload a task; an example file can be found inside `examples/sample-ranking-task.xml` within the Appraise package.

7. Experiments

7.1. Appraise in EuroMatrixPlus

As mentioned earlier in this article, we have created Appraise to support research work on hybrid machine translation, especially during the EuroMatrixPlus project. This is described in (Federmann et al., 2009, 2010; Federmann and Hunsicker, 2011; Hunsicker et al., 2012).

7.2. Appraise in taraXÜ

We have also used Appraise in the taraXÜ project, conducting several large annotation campaigns involving professional translators and language service providers. Results from this research work are summarised in (Avramidis et al., 2012).

7.3. Appraise in T4ME

In the T4ME project, we investigate how hybrid machine translation can be changed towards optimal selection from the given candidate translations. Part of the experimental setup is a shared task (ML4HMT) in which participants have to implement this optimal choice step. We used Appraise to assess the translation quality of the resulting systems. This is described in (Federmann, 2011; Federmann et al., 2012a,b).

Appraise has also been used in research related to the creation of standalone hybrid machine translation approaches. Related work is published as (Federmann, 2012).

7.4. Appraise in MONNET

We also used Appraise in the context of terminology translation for the business domain. These experiments are conducted as part of the MONNET project and are presented in (Arcan et al., 2012).

8. Conclusion and Outlook

We have described Appraise, an open-source tool for manual evaluation of machine translation output, implementing various annotation tasks such as ranking or error classification. We provided detailed instructions on the installation and setup of the tool and gave some brief introduction to its usage. Also, we reported on research work for which different versions of Appraise have been used, feeding back into the tool's development.

Maintenance and development efforts of the Appraise software package are ongoing. By publicly releasing the tool on GitHub, we hope to attract both new users and new developers to further extend and improve it. Future modifications will focus on new annotation tasks and a more accessible administration structure for large numbers of tasks. Last but not least, we intend to incorporate detailed visualisation of annotation results into Appraise.

Acknowledgements

This work has been funded under the Seventh Framework Programme for Research and Technological Development of the European Commission (grant agreement no.: 249119) through the T4ME contract. We are grateful to the anonymous reviewers for their valuable feedback on the draft version of this paper.

```
<set id=”spiegel-20120210” source-language=”ger” target-language=”eng”>
    <seg id=”1” doc-id=”source-text.de.txt”>
      <source>In der syrischen Stadt Aleppo sind nach staatlichen Angaben
           mehrere grosse Sprengsätze detoniert, offenbar vor zwei Einrichtungen
            der Sicherheitskräfte.</source>
      <translation system=”google”>In the Syrian city of Aleppo after
           government data several large bombs are detonated, apparently, two
           institutions of the security forces.</translation>
      <translation system=”bing”>In Aleppo, Syria, Syrian several large
           explosive devices are detonates according to State, apparently before
            two installations of the security forces.</translation>
      <translation system=”yahoo”>In the Syrian city Aleppo detonated according
            to national instructions several large explosive devices, obviously
           before two mechanisms of the security forces.</translation>
    </seg>
    ...
</set>
```

Figure 5. Excerpt of sample import XML for an Appraise ranking task. For consistency and ease of use, the same format is used for all annotation tasks. The full file is available as `examples/sample-ranking-task.xml` *from the Appraise software package.*

Bibliography

Arcan, Mihael, Christian Federmann, and Paul Buitelaar. Using Domain-specific and Collaborative Resources for Term Translation. In *In Proceedings of the Sixth workshop on Syntax, Structure and Semantics in Statistical Translation*, Jeju, South Korea, July 2012. Association for Computational Linguistics (ACL).

Avramidis, Eleftherios, Aljoscha Burchardt, Christian Federmann, Maja Popovic, Cindy Tscherwinka, and David Vilar Torres. Involving Language Professionals in the Evaluation of Machine Translation. In *8th ELRA Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), 2012.

Bennett, E. M., R. Alpert, and A. C. Goldstein. Communications Through Limited-response Questioning. *Public Opinion Quarterly*, 18(3):303–308, 1954. doi: 10.1086/266520.

Bird, Steven, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009. ISBN 978-0-596-51649-9. doi: http://my.safaribooksonline.com/9780596516499. URL `http://www.nltk.org/book`.

Bojar, Ondrej, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W11-2101`.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on*

Statistical Machine Translation, pages 70–106, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W08/W08-0309.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia, editors. *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Montréal, Canada, June 2012. URL http://www.aclweb.org/anthology/W12-31.

Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. ISSN 0013-1644.

Denkowski, Michael and Alon Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology-new/W/W11/W11-2107.

Federmann, Christian. Appraise: An Open-Source Toolkit for Manual Phrase-Based Evaluation of Translations. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valetta, Malta, May 2010. URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/197_Paper.pdf.

Federmann, Christian. Results from the ML4HMT Shared Task on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid Machine Translation. In *Proceedings of the International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT 2011) and of the Shared Task on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid Machine Translation (ML4*. META-NET, 11 2011.

Federmann, Christian. Can Machine Learning Algorithms Improve Phrase Selection in Hybrid Machine Translation? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–118. Association for Computational Linguistics (ACL), European Chapter of the Association for Computational Linguistics (EACL), 4 2012.

Federmann, Christian and Sabine Hunsicker. Stochastic Parse Tree Selection for an Existing RBMT System. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 351–357, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2141.

Federmann, Christian, Silke Theison, Andreas Eisele, Hans Uszkoreit, Yu Chen, Michael Jellinghaus, and Sabine Hunsicker. Translation Combination using Factored Word Substitution. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 70–74, Athens, Greece, March 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W09/W09-0x11.

Federmann, Christian, Andreas Eisele, Yu Chen, Sabine Hunsicker, Jia Xu, and Hans Uszkoreit. Further Experiments with Shallow Hybrid MT Systems. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 77–81, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-1708.

Federmann, Christian, Eleftherios Avramidis, Marta R. Costa-jussa, Josef van Genabith, Maite Melero, and Pavel Pecina. The ML4HMT Workshop on Optimising the Division of Labour in Hybrid Machine Translation. In *8th ELRA Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), 5 2012a.

Federmann, Christian, Maite Melero, and Josef van Genabith. Towards Optimal Choice Selection for Improved Hybrid Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 97:5–22, 4 2012b.

Fleiss, J.L. Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, 76 (5):378–382, 1971.

Hunsicker, Sabine, Yu Chen, and Christian Federmann. Machine Learning for Hybrid Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 312–316, Montréal, Canada, June 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W12-3138`.

Krippendorff, Klaus. Reliability in Content Analysis. Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433, 2004.

Och, Franz Josef. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075117.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL `http://acl.ldc.upenn.edu/P/P02/P02-1040.pdf`.

Scott, William A. Reliability of Content Analysis: The Case of Nominal Scale Coding. *The Public Opinion Quarterly*, 19(3):321–325, 1955.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. Error Analysis of Machine Translation Output. In *International Conference on Language Resources and Evaluation*, pages 697–702, Genoa, Italy, may 2006.

Address for correspondence:
Christian Federmann
`cfedermann@dfki.de`
DFKI Gmbh—Language Technology Lab
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany