CleanEval: a competition for cleaning webpages

Marco Baroni[⋆], Francis Chantree[†], Adam Kilgarriff[†], Serge Sharoff[‡]

University of Trento[⋆], Lexical Computing Ltd[†], University of Leeds[‡]

Abstract

Cleaneval is a shared task and competitive evaluation on the topic of cleaning arbitrary web pages, with the goal of preparing web data for use as a corpus for linguistic and language technology research and development. The first exercise took place in 2007. We describe how it was set up, results, and lessons learnt.

1. Introduction

More and more language technology research and development uses the web as its data source (Baroni and Bernardini, 2006; Fairon et al., 2007; Hundt et al., 2007; Kilgarriff and Grefenstette, 2003). The following questions always arise:

1. how do we detect and get rid of "boilerplate" - that is: navigation bars, headers, footers and other textual data of no linguistic interest
2. how do we identify paragraphs and other structural information
3. how do we produce output in a standard form of regular text suitable for further linguistic processing?

It is a low-level, unglamorous task and yet it is increasingly crucial: the better it is done, the better the outcomes. All further layers of linguistic processing depend on the cleanliness of the data. If we use a web-corpus with uncleaned data, the most significant bigrams will often be *Click here* or *Further information*. This distorts the language model considerably.

To date, cleaning has been done in isolation by each group using web data (and it has not been seen as interesting enough to publish on). Resources have not been pooled, and it has often not been done well. In Cleaneval we put cleaning centre-stage. The goals of the exercise are to identify good cleaning strategies and to foster sharing of ideas and programs.

Cleaneval takes the form of an open competition: who can do the best job of cleaning arbitrary web pages? It may seem odd to foster collective effort through competition, but the evidence from a number of competitions (Senseval, NIST Open MT, ACE, etc.) is that it works: the process of setting up the exercise precipitates discussion about the critical questions in the field, the 'game' aspect brings in additional participants including junior ones who might otherwise find it hard to gain entry into the field, and the whole exercise sets benchmarks for the field which then become common reference points. It also supports progress, with the field as a whole benefiting from the leading technologies as identified in the competition, and because future rounds of the exercise can build on previous ones in iterations of the cycle. For discussions of the approach and its benefits, see, e.g., Belz and Kilgarriff (2006), Gaizauskas (1998).

The stages of the process are:

1. Announce the overall theme of the evaluation and invite people to participate
2. Identify data; divide between development set and test set
3. Employ people to produce sets of correct answers (the "gold standard")
4. Distribute development set (with correct answers)
5. Develop scoring software (the "scorer")
6. Distribute test data (without correct answers)
7. Participants process data, submit their system's answers
8. Organisers score participants' systems
9. Workshop

A first Cleaneval exercise was held in summer 2007 under the auspices of ACL's Special Interest Group on Web as Corpus, with workshop in September.[1] We addressed two languages, English and Chinese: English, because it is the largest and most important on the web, and Chinese, firstly, as evidence that we were not blindly anglocentric, and secondly, to explore the problems that a language with a variety of competing character sets and challenging tokenization issues might present. In this paper we describe the preparation of the data, scoring, and results. For descriptions of participating systems see individual papers in Fairon et al. (2007).

2. Data preparation

2.1. Data selection

The basic unit was the web page. For the exercise we used a random sample of pages from Web corpora that had already been developed for English and Chinese as described in Sharoff (2006). Thus the data samples carry the imprint of the choices made in the development of those corpora, for example the method of page collection and exclusion of pages that were too long, or too short, or presented *prima facie* evidence of not containing usable text for a text-centred corpus. The corpora were collected from URLs returned by making queries to Google, which consisted of four words frequent in an individual language. We have previously established that if mid-frequency words like *picture, extent, raised* and *events* are all used in a query, retrieved pages are likely to contain extended stretches of text (Sharoff, 2006).

[1] The workshop was joint with WAC3, the third Web-as-Corpus workshop.

	Dev EN	Dev ZH	Test EN	Test ZH
Files	57	60	684	653
KBytes	1892	1943	10701	9845

Table 1: Data sets

While the method for data selection is open to challenge, cleaning techniques will always be applied to pages which are the output of corpus-collection strategies, and the corpus-collection strategies behind the corpora we used are documented and reasonably generic.

For Cleaneval-1 we used only html pages. We acknowledge the issues involved in gathering and cleaning PDF or Word files, and anticipate that they will feature in future Cleanevals, but, given the small scale and short time frame for Cleaneval-1 we chose to leave them out of this first exercise.

We divided the data into a development set and an evaluation set, whose sizes are reported in Table 1.

In most shared task exercises, the ratio of training to evaluation data has been higher. This is typically because the organisers want to support supervised machine-learning methods. We imagined that these kinds of methods would not be the most suitable for this task, as there are so many different varieties of 'dirt' to be cleaned. However, at the event several systems did use ML methods and, for the next exercise, it is likely that effort will be put into preparing a large training set.

2.2. Annotation guidelines

The annotators were instructed as follows:

> Your task is to "clean up" a set of webpages so that their contents can be easily used for further linguistic processing and analysis. In short, this implies
>
> 1. removing all HTML/JavaScript code and "boilerplate" (headers, copyright notices, link lists, materials repeated across most pages of a site, etc.);
> 2. adding a basic encoding of the structure of the page using a minimal set of symbols to mark the beginning of headers, paragraphs and list elements. ...

This is the opening of the annotation guidelines (available at `http://cleaneval.sigwac.org.uk/annotation_guidelines.html`); the guidelines comprise two pages, and include examples. Given that this was the first exercise of its kind and time was limited, we chose to trust the annotators to be able to follow these instructions in a systematic and replicable way, rather than legislating in the guidelines for many different cases.

All files in the development set were annotated by two people, to test how reliable the identification of boilerplate can be. Most of the time the decisions made by each annotator were identical. Where there were differences, it was often because some annotators preferred to err on the side of caution by retaining text such as the date when a discussion was published, or links to other relevant pages, while others deleted it, considering it boilerplate. The average pair-wise score (computed as described in Section 3. below) for files manually cleaned by different annotators was 94%, substantially higher than the best systems taking part in the competition.

2.3. Annotation setup and output document format

We recruited 23 Masters students in Computer-Assisted Translation (CAT) at the University of Leeds, including some Chinese native speakers who worked on the Chinese data, while all participants had a sufficient level of English to work on English. All annotators were familiar with html tags as they had dealt with them with them when using CAT packages such as Trados or Wordfast. Headers, navigation bars etc. require special attention in translation (since they are translated differently from running text); so we expected these annotators to be well-suited for the manual markup task.

The annotators worked with two windows open, one showing the page as rendered by a browser, and the other showing a pre-cleaned version of the page, in a plain-text editor (by default NotePad++).[2] The pre-cleaning was done with a simple script that removed html markup, JavaScript and other clearly unwanted material. It also converted all pages to UTF-8. (`enca` was used for automatic conversion of Chinese encodings.) The script was made available to participants.

The output produced by the annotator had all boilerplate removed and simple markup added. The markup to be added was limited to opening tags *p, h, l* for paragraphs, headers and lists. In order to keep the framework very simple, we assumed no nesting of tags, with every tag implicitly closing the currently-open element. A simplified example of original input, automated stripping and the final output is presented in Figure 1.

At the workshop there were criticisms that our simple format implied that the link between the original markup and the retained material was lost. There are numerous reasons for keeping the original markup in the corpus: it provides features for machine learning of the cleaning task, or for genre detection (Santini, 2007), and it is of central importance if the goal is to study the graph structure of websites. We accept the criticism. At the time we needed to set up arrangements that allowed the annotators to start work promptly, on dependable software: there was neither the time nor the money to set up an editor that retained the html markup in a way that did not make the editing more cumbersome for the annotators. Our purpose was to generate corpora from the web that can be used as linguistic materials, and it was not evident that the source markup was relevant for that task.

The annotators cleaned an average of 120 kB per hour for English and 50kB per hour for Chinese (and were paid a

[2] `http://notepad-plus.sourceforge.net/`

```
<a href="http://www.environment-agency.wales.gov.uk/">
<img src="/common/images/toolbar/banner_index_home_off.gif" width=47
height=24 hspace=0 vspace=0 border=0 alt="Home" align="left"
name="banner_index_home"></a>
<a href="http://www.environment-agency.gov.uk/news/?lang=_e">
<img src="/common/images/toolbar/banner_index_news_off.gif" width=40
height=24 hspace=0 vspace=0 border=0 alt="News" align="left"
name="banner_index_news"></a>
<h3><font face="Arial"><a name="eutro"></a>Eutrophication</font></h3>
<p><font face="Arial" size="2">Concentrations in Welsh rivers of the main
plant nutrients (phosphate and nitrate) are generally much lower than those
found in the midlands and south-east England.</font></p>
```

```
Home News

Eutrophication

Concentrations in Welsh rivers of the main plant nutrients (phosphate
and nitrate) are generally much lower than those found in the midlands
and south-east England.
```

```
<h>Eutrophication

<p>Concentrations in Welsh rivers of the main plant nutrients (phosphate
and nitrate) are generally much lower than those found in the midlands
and south-east England.
```

Figure 1: Original page; after automatic HTML stripping; after manual cleaning/markup.

rate per kB of data cleaned). The process of annotation was completed in March 2007. It was funded by a small grant from Lexical Computing Ltd. The funding available defined the quantity of annotation that could take place. We prioritised "quantity of data" over double-annotation, and, as noted above, we only double-annotated the development sets.

3. Scoring

Our scorer needed to measure the similarity between two differently cleaned versions of a file. For the actual Cleaneval evaluation these would be a participant's version and the gold standard version. For the development of the scorer, we did not have participants' versions available, but we did have pairs of versions of the same file cleaned by different annotators. For a number of these pairs, two of the co-authors provided assessments of whether the pair were similar to each other or not, and this provided training data to identify which features and parameters we should use.
The task has two aspects: removal of boilerplate, and insertion of *p*, *h* and *l* tags. The scoring framework needed to review both aspects, and, if we were to give a single score, their relative importance.

3.1. Scoring Metrics

The primary method of scoring was *Levenshtein edit distance*, which measures the distance between two strings given by the minimum number of *operations* – insertions, deletions, or substitutions of a single character – needed to transform one into the other. We adapted the metric by substituting 'token' (typically a word) for 'character'; we also did not allow substitutions, as insertion of a wrong token for the correct one is in our case not allowable as a single mistake. Using this metric we calculated *misalignment* between each pair of cleaned files. This is the edit distance divided by the file length, i.e. the percentage of all tokens from either of the two files that cannot be matched with a token in the other file.

The algorithm gave a perfect measurement of minimum edit distance, as it built matrices of the entire files and found the paths of minimum cost through them. As a result of this, though, it was slow to run.

We also wished to investigate *granularity*, a measure of how clumped or dispersed (respectively, low or high granularity) the differences are between two differently cleaned files. The former happens because sequences of misalignments result from much fewer actual differences of opinion – represented by the initial misalignments of the sequences. The latter happens when the differences of opinion approach the number of misaligned words. We entertained two contradictory intuitions about how granularity may be indicative of good cleaning: low granularity represents greater agreement, but high granularity may represent occasional and possibly inconsequential differences.

To assess solely the markup which had been added in to the text, we computed *segment validity*. We define this as being the similarity between two files in terms of their structure, as indicated by matching up of segments between the two. Insertions of paragraph, header and list tags are the triggers to identifying these segments.

Prior to running our scorer using these metrics, we carried out some standard preprocessing on all files. We normalised whitespace, removed blank lines and tokenised the text. We normalised files further by lower-casing, and deleting punctuation and other non-alphanumeric characters, though we also experimented with omitting this process. Tokenising proved to be a problem when building a Chinese version of the scoring program as there are no explicit word boundaries. In the end our Chinese scorer relied on tags and newlines.

3.2. Training the Scorer

We trained the scorer to agree maximally with human expert judgments. We took pairs of differently-cleaned versions of the same original files and two of the co-authors manually classifying them according to whether the first version was cleaned better, worse, or about the same as the second. A variety of scoring parameters were identified, all considered potentially important in determining cleaning success. We then trained our scorer by adjusting these parameters to best predict (i.e. replicate) the human classifications. It is in this comparison against expert judgment that our scorer captures to some extent what is meant by good cleaning.

A selection of the scorer's training runs are shown in Table 2 and Table 3, for edit distance-based and segment validity calculations respectively. In both tables, *error rate* is the percentage by which the similarity between two versions of a file predicted by the scorer differs from the similarity between them ascribed by the human experts. For calculations based on edit distance this difference is the misalignment of the files; for segment validity it is solely the mis-matching of the segments of the files. For the former, all tokens are considered; for the latter, only the inserted markup tags and a few adjacent tokens to facilitate the segment identification process. Our objective here was to adjust the parameters to minimise the error rates, thereby representing maximal prediction of the human classifications.

Granularity is calculated as the total number of changes in operations made by the scorer, as a percentage of the edit distance. Repeated *similar* operations – contiguous tokens that appear in one version but not in the other – indicate clumping and therefore low granularity; conversely, *switching* operations – a token in the first file not in the second, followed by one in the second that's not in the first – adds to granularity. Our objective was to see if granularity predicted the error rate; if so, we would use it as a parameter.

The following example shows two differently-cleaned files, one with low granularity and with high granularity, both with an edit distance of four, compared against a gold standard version, and the original website text:

Run	Normalised?	Generalise Tags?	Remove Markup?	Error Rate	Granularity
ED1	Y	N	N	31.3	58.1
ED2	N	N	N	31.3	58.0
ED3	Y	Y	N	29.9	25.6
ED4	N	N	Y	22.9	18.8

Table 2: Scoring parameters: edit distance and granularity

Run	Tag Type	Tags Begin or End Segment?	Number of Offset Lines (N)	Error Rate
SV1	P	B	2	23.6
SV2	P	E	2	17.4
SV3	H	B	2	26.4
SV4	H	E	2	18.1
SV5	L	B	2	28.5
SV6	L	E	2	28.5
SV7	Gen	B	2	22.9
SV8	Gen	E	2	17.4

Table 3: Scoring parameters: segment validity

original text:	a		c	d	e		h	i	j	k	l	m
gold standard:	a		c	d	e	f				k	l	m
low gran. (=2/4):			c	d	e	f		h	i	j	k	l	m
high gran.(=5/4):	a	b	c	d	e		g			j	k		m

As can be seen, the markup tags entered by the contestants (e.g. `b`, `f` and `g` in the example above) can have a large effect on both edit distance and granularity.

Segment validity is calculated individually for each type of segment, as determined by the contestant-entered tags: paragraph, header, list item, and generalised (when all tags are renamed as one generic type). We investigated whether a tag successfully marked either a segment's beginning or its end, with correct alignment of a pre-determined number of *offset lines*, respectively before or after the tag, being proof of segment validity. Thus, if the same tag appears in two files and the same number of offset lines also match, then that segment is potentially a valid match between that pair of files.

Training the scorer by varying the edit distance parameters reveals that:

- generalising markup tags makes prediction easier (ED3 vs. ED1)
- removing markup tags makes prediction even more easy (ED4 vs. ED1)
- normalising has little or no effect on the outcome (ED2 vs. ED1)
- higher granularity tends to indicate greater discrepancy (all runs shown here); however, this was found to be largely due to the introduction of the markup tags

Training the scorer by varying the segment validity parameters reveals that:

50%: Alignment without markup
50%: Alignment with markup, comprising:
25%: Edit Distance
25%: Segment Validity, comprising:
4.17%: <p> as segment beginner
4.17%: <p> as segment ender
4.17%: <h> as segment beginner
4.17%: <h> as segment ender
4.17%: <l> as segment beginner
4.17%: <l> as segment ender

Table 4: Scoring parameters: final weightings

- 'paragraph' was the most consistently added tag, followed by 'header', with 'list item' the least (SV1-6)
- generalising the tags made prediction easier (SV7,8 vs. the others)
- it is more predictive to consider that tags end segments rather than begin them (SV2,4,6,8 vs. SV1,3,5,7)
- (Not shown here: very little difference was found when N>2; making N=1 helps alignment of 'header' and 'list item' tags but not 'paragraph' tags.)

3.3. Scorer Conclusions and Finalisation

The conclusions we draw, and which we implemented as the parameters of the scorer in the Cleaneval exercise, were:

- contestant-inserted tags introduce more disagreement between taggers than boilerplate-removal does: we consider that good cleaning may be of more importance than good tag insertion, so we run the scorer without markup as well as with it
- N=2 for paragraphs, N=1 for headers and list items
- the results of the granularity metric are inconclusive: we do not use it
- normalisation: it does no harm and although it did not help on our training set it might have greater influence on others; it was retained

The weightings that we gave to the various factors in the final version of the scorer used in the Cleaneval exercise are shown in Table 4.

4. Participants and Results

For Chinese only one participating system, from University of Osnabrück, returned results in the format suitable for the script (other submissions had problems with either encodings or file format). The system performance was 18%, although it is likely that the low figure results from the scoring algorithm not aligning on appropriate units: this was not resolved by the time of the workshop.

For English there were nine participants, from four continents and from both academia and one company. The participants and their results are shown in Table 2, in terms of "Text and Markup" (TM) , "Text-Only" (TO) and the Average (Ave) of the two methods.

The results are remarkably close. Except for two student outliers, all 'average' results are between 70% and 75%, with the four highest-scoring systems all between 74% and 75%. Adding the markup correctly was substantially harder than simply finding which text to retain, as shown by TO scores being around 20% higher than TM results.

5. Discussion, lessons learnt, and way forward

We have completed a first run of the Cleaneval exercise. It ran satisfactorily. Several points emerged in the discussions at the workshop. Contrary to our expectations, participants were largely interested in using supervised machine learning techniques, for which a larger training set is required. Also, most systems used the HTML structure of the input page as an input to their algorithms, so it would have been preferable to annotate the gold standard with markup of where the "good text" begins and ends within the original document, rather than providing cleaned pages with all the "bad text" removed.

Inserting markup was more problematic than removing boilerplate.

There was a high level of interest on there being a further exercise; this should include more languages and should cover pdf as well as html documents. It should also address POS-tagging: POS taggers which have not been developed on web data typically perform badly on them, so it is interesting to add another (optional) task to the exercise which aims to support the development of POS-taggers which perform well on web data. A volunteer to run the next exercise was also discovered.

Acknowledgments

Thanks to Tony Hartley and Mike Higgins, University of Leeds, for helping us to arrange cleaning and annotation, and to Pavel Rychly for his assistance with running the evaluation.

6. References

Baroni, M. & Bernardini, S. (Eds.) (2006). *Wacky! Working papers on the Web as Corpus*. Bologna: Gedit.

Belz, A. & A. Kilgarriff. (2006). Shared-task evaluations in HLT: Lessons for NLG. *Proceedings of the 4th International Conference on Natural Language Generation (INLG'06)*, 133-135

Fairon, C., Naets, H., Kilgarriff, A. & de Schryver, G.-M. (Eds.). (2007). *Building and exploring web corpora – Proceedings of the 3rd Web as Corpus Workshop, incorporating Cleaneval.* Louvain: Presses Universitaires de Louvain.

Gaizauskas, R. (1998). *Evaluation in language and speech technology Journal of Computer Speech and Language* 12(3), 249-262.

Hundt, M., Nesselhauf, N. & Biewer, C. (Eds.). (2007). *Corpus linguistics and the web*. Amsterdam: Rodopi.

Kilgarriff, A. & Baroni, M. (Eds.). (2006). *Proceedings of the 2nd International Workshop on the Web as Corpus.* East Stroudsburg (PA): ACL.

Students	**TM**	**TO**	**Ave**	**Non-students**	**TM**	**TO**	**Ave**
Bauer et al, Osnabrück Uni, Germany	53.5	73.5	63.5	Gao & Abou-Assaleh, GenieKnows, Canada	63.9	83.4	73.6
Marek, Pecina & Sprousta; Charles Uni, Czech Republic	65.3	**84.1**	**74.7**	Girardi, FBK, Italy	**65.6**	82.5	74.0
Hofmann and Weerkamp, Uni Amsterdam, NL	65.5	83.0	74.2	Saralegi & Leturia, Elhuyar Foundation, Spain	65.3	83.4	74.3
Chaudhury, India	59.5	80.9	70.2	Evert, Osnabrück Uni, Germany	60.3	82.9	71.6
Conradie, North West Uni, South Africa	45.5	60.2	52.9				

Table 5: Participants and results

Kilgarriff, A., Grefenstette, G. (2003). Introduction to the special issue on the Web as Corpus. *Computational Linguistics*, 29(3), 333–347.

Santini, M. (2007). *Automatic identification of genre in Web pages*. Ph.D. thesis, University of Brighton.

Sharoff, S. Creating general-purpose corpora using automated search engine queries. In: Baroni and Bernardini (2006), pp. 63-98.