The Good, the Bad, and the Hazy: Design Decisions in Web Corpus Construction

Roland Schäfer
Freie Universität Berlin
roland.schaefer@fu-berlin.de

Adrien Barbaresi
Freie Universität Berlin
adrien.barbaresi@ens-lyon.fr

Felix Bildhauer
Freie Universität Berlin
felix.bildhauer@fu-berlin.de

Abstract

In this paper, we examine notions of text quality in the context of web corpus construction. Web documents often contain material which disqualifies them from inclusion in a corpus (tag clouds, lists of names or nouns, etc.). First, we look at the agreement between coders (especially corpus designers) given the task of rating text quality. Then, we evaluate a simple and fully unsupervised method of text quality assessment based on short and very frequent words. Finally, we describe our general approach to the construction of carefully cleansed and non-destructively normalized web corpora. Under this approach, we annotate documents with quality metrics instead of actually removing those documents classified as being of low quality.

1 Introduction

1.1 The Text Criterion

Crawled raw data for web corpus construction contains a lot of documents which are technically in the target language, but which fail as a text. Documents just containing tag clouds, lists of names or products, etc., need to be removed or at least marked as suspicious. Defining the criteria by which the decision to remove a document is made, however, is quite difficult. For instance, many documents contain a mix of good and bad segments and thus represent borderline cases. The decision to systematically remove documents is thus a design decision with major consequences for the composition of the corpus and with potential negative side effects on the distribution of linguistic features. Certain linguistic phenomena might be more or less accidentally underrepresented (w. r. t. the population and/or some specific design criteria) if very long or very short documents are not included, for example. On the other hand, certain lemmas or parts-of-speech might be overrepresented if long word lists or lists of names are not removed, etc. Therefore, while this paper raises mostly technical questions which corpus designers have to care about, we are convinced that linguists working with web corpora should also be aware of how such technical matters have been dealt with.

We first examine how well humans perform given the task of classifying documents as good or bad web corpus documents (Section 2). Then, we introduce and evaluate a completely unsupervised method to classify documents according to a simple but effective metric (Section 3). Finally, we introduce a format for the representation of corpora in which cleanups like boilerplate detection and text quality assessment are not actually executed as deletion. Instead, we keep the potentially bad material and mark it as such (Section 4).

1.2 Context of the Experiment

The work presented here was carried out as part of the construction of the COW2013 corpora, improved versions of the COW2012 corpora (Schäfer and Bildhauer, 2012).[1] The corpora, available in various languages, are all of giga-token (GT) size.[2] Our design goals and the usage scenarios for our web corpora do not allow us to create corpora which are just bags of (very clean) sentences in random order like, for example, the corpora in the Leipzig Corpora Collection (Biemann et al., 2007).[3] We keep whole documents and are generally very careful with all cleanup and normalization steps, simply because the line between noise and corpus material is often difficult to draw. Also, there are many areas of (computational) linguistics

[1] http://www.corporafromtheweb.org/

[2] Currently: Danish 1.5 GT (estimate), Dutch 3.4 GT, English 6 GT (estimate), French 4 GT (estimate), German 9.1 GT, Spanish 1.6 GT, Swedish 2.3 GT.

[3] Cf. also Biemann et al. (2013) for a discussion of different tool chains and their implementation.

for which single sentences are insufficient, such as (web) genre research, information structure, variants of distributional semantics, and even syntax which deals with effects which go beyond single sentences (e. g., the syntax of sentence connectors). Furthermore, one of our future plans is to take uniform random samples from the web by advanced crawling algorithms in order to build small but highly representative web corpora for linguistic web characterization.[4] Although we will always require corpus documents to fulfill minimal linguistically motivated criteria, this general empirically motivated sampling approach does not allow us to filter documents and sentences aggressively, as it would be possible in many more task-oriented settings.

2 Rating Text Quality

2.1 Data Set and Task

Our primary goal in this study was to find out whether corpus designers have clear intuitions about the text quality of web documents, and whether they could operationalize them in a way such that others can reproduce the decisions. Therefore, we randomly selected 1,000 documents from a large breadth-first crawl of the `.uk` TLD executed with `Heritrix` (Mohr et al., 2004).[5] It is the crawl which serves as the basis for our UKCOW2012 and UKCOW2013 corpora. The first 500 documents of the sample were from the initial phase of the crawl, the second 500 from the final phase (after eight days of crawling), when the average quality of the documents is usually much lower (shorter documents, web shops, etc.).[6] The documents were pre-processed with the `texrex` software for HTML stripping, boilerplate removal, code page normalization, etc., and were thus reduced to plain text with paragraph boundaries.[7]

Then, three coders (A, R, S) were given the task of rating each document on a 5-point scale $[-2..2]$ as to how good a corpus document it is.[8] Coders A and R were corpus designers (the second and first author of this paper) with a shared understanding of what kind of corpus they want to build. Coder S was a student assistant who had previously participated in at least three related but not identical rating tasks on the same kind of data, amounting to at least five work days of coding experience.

A series of criteria was agreed upon, the most important being:

- Documents containing predominantly full sentences are good, "predominantly" meaning considerably more than 50% of the text mass (as perceived by the coder).
- Boilerplate material in sentence form is good (*You are not allowed to post comments in this forum.)*, other boilerplate material is bad (*Copyright © 2046 UAC Ltd.*).
- Sentences truncated or otherwise destroyed by some post-processing method are good as long as they are recognizable as (the rest of) a sentence.
- Repetitions of good sentences are good.
- Decisions should not depend on the length of the document, such that a document containing only one good sentence would still be maximally good.
- Non-English material contributes to badness.
- Non-sentence material (lists, tables, tag clouds) contributes to badness.
- However, if a list etc. is embedded in a coherent text which dominates the document, the document is good (prototypically recipes with a substantial amount of instructions).

The scale is interpreted such that 1 and 2 are assigned to documents which should definitely be included in the corpus, -1 and -2 to documents which should not be included, and 0 to borderline cases. In an initial phase, the coders coded and discussed one hundred documents together (which were not included in the final sample) to make results more consistent.[9]

2.2 Results

Table 1 summarizes the results. Despite clear guidelines and the initial training phase, the best

[4] To our knowledge, this has not been done so far. Cf. Chapter 2 of Schäfer and Bildhauer (2013) for an introduction to the problems of uniform sampling from the web and to web characterization. Relevant original papers include Henzinger et al. (2000) and Rusmevichientong et al. (2001).

[5] The data set and the coder data described below can be obtained from the first author.

[6] We will refer to the two subsamples as "early data" and "late data" from now on.

[7] `http://sourceforge.net/projects/texrex/`

[8] There are of course no intrinsically bad or good documents. What we try to measure is the "textiness" of documents, using "goodness" and "badness" as abbreviations for "textiness" and "non-textiness".

[9] It was found in a meta analysis of coder agreement in computational linguistics tasks (Bayerl and Paul, 2011) that training is a crucial factor in improving agreement.

statistic	early 500	late 500	all 1,000
raw	0.566	0.300	**0.433**
κ (raw)	0.397	0.303	0.367
$ICC(C,1)$	0.756	0.679	**0.725**
raw ($r \geq 0$)	0.900	0.762	**0.831**
raw ($r \geq 1$)	0.820	0.674	0.747
κ ($r \geq 0$)	0.673	0.625	**0.660**
κ ($r \geq 1$)	0.585	0.555	0.598
κ ($r \geq 2$)	0.546	0.354	0.498

Table 1: Inter-coder agreement for the text quality rating for 1,000 web documents by three coders; below the line are the results for ratings converted to binary decisions, where $r \geq n$ mean that any rating $r \geq n$ was counted as a positive decision; κ is Fleiss' Kappa and ICC the intraclass correlation.

value ($ICC = 0.756$) on the early 500 documents is mediocre. When the documents get worse in general (and also shorter), the confusion rises ($ICC = 0.679$). Notice also the sharp drop in raw agreement from 0.566 to 0.300 between the early and the late data.

Since Fleiss' κ is not very informative on ordinal data and the ICC is rarely reported in the computational linguistics literature, we also converted the coders' ordinal decisions to binary decisions at thresholds of 0, 1, and 2.[10] The best value is achieved with a threshold of 0, but it is below mediocre: $\kappa = 0.660$ for the whole data set. The value is in fact below the interval suggested in Krippendorff (1980) as acceptable. Even if Krippendorff's interval $(0.67, 0.8)$ is not the final (task-independent) word on acceptable κ values as suggested, for example, in Carletta (1996) an Bayerl and Paul (2011), then 0.660 is still uncomfortably low for the creation of a gold standard. For the binary decisions, the raw agreement also drops sharply from 0.900 to 0.762 between the early and the late material.

It should be noted that coders judge most documents to be quite acceptable. At a threshold ≥ 0 on the 5-point scale, coder A considers 78.4% good, coder R 73.8%, and coder S 84.9%. Still, there is an 11.1% difference between R and S. Positive decisions by R are almost a perfect subset of those by S, however. In total, 73.0% are rated as good by both coders.

We would like to point out that one of the crucial results of this experiment is that corpus designers themselves disagree substantially. Surely, it would be possible to modify and clarify the guidelines, do more training, etc.[11] This would most likely result in higher inter-coder agreement, but it would mean that we operationalize a difficult design decision in one specific way. It has been shown for similar tasks like boilerplate classification that higher inter-coder agreement is possible (Steger and Stemle, 2005). If, however, paragraphs and documents are deleted from the corpus, then users have to agree with the corpus designers on the operationalization of the relevant decisions, or they have to look for different corpora. Our approach is attempt to remedy this situation.

3 Text Badness as the Lack of Function Words

3.1 Summary of the Method

We suggest to use a single criterion in an unsupervised approach to document quality assessment, based on ideas from language identification. In addition to being unsupervised, the approach has the advantage of allowing for very time-efficient implementations. Although the proposed method is arbitrary to a certain degree, it is not a heuristic in the proper sense. As we are going to show, results are quite consistent. Furthermore, considering the degree of arbitrariness involved in human decisions about document quality, we argue against rigorous corpus cleaning and normalization (given the aims and usage scenarios described in Section 1.2) and for non-destructive normalization.

Most approaches to language identification following early papers like Cavnar and Trenkle (1994) and Dunning (1994) use character n-gram statistics. An alternative using short and frequent words is described in Grefenstette (1995). This method (also called the dictionary method) has not been used as prominently as the character n-gram method, but some recent approaches also apply it in the context of normal language identification, e. g., Řehůřek and Kolkus (2009).

[10] Some readers could object that it would have been better to let coders make binary decisions in the first place or redo the experiment in such a way. However, we designed the task specifically because in our earlier informal evaluations and discussions, we had noticed the substantial amount of borderline cases. Using binary decisions or any scale without a middle option would not have captured the degree of undecidability equally well.

[11] Even the word "training" is problematic here, because it is unclear who should train whom.

Clearly, the short word method bears some potential also for text quality detection, because a low frequency of short and frequent words (mostly function words) is typical of non-connected text such as tag clouds, name lists, etc.[12] For the WaCky corpora (Baroni et al., 2009), precompiled lists of words were used, combined with thresholds specifying the required number of types and tokens from these lists in a good document. In Schäfer and Bildhauer (2012), our similar but completely unsupervised method was suggested. It must be mentioned that it only works in an unsupervised manner for web corpora from TLDs with one dominant language. In more complicated scenarios (multilingual TLDs or non-scoped crawls), it has to be combined with normal (i. e., character n-gram based) language identification to pre-filter training documents.[13]

In the training phase, the *n* most frequent word types are calculated based on a sample of documents from the corpus. For each of these types, the weighted mean of its relative frequency in the sampled documents and the corresponding weighted standard deviation are calculated (weighted by the length of the document) as an estimate of the corpus mean and standard deviation. In the production run, these two statistics are used to calculate the normalized deviation of the relative frequency of these *n* types in each corpus document. The more the frequency in the document deviates negatively from the estimated population mean, the worse the document is assumed to be. If the added normalized negative deviation of the *n* types (the "Badness" of the document) reaches a threshold, the document is removed from the corpus. Both in the training and the production run, documents are processed after markup stripping and boilerplate removal.

In practice, we log-transform the relative frequency values because this gave us more consistent results in the initial evaluation. Also, the component value contributed by each of the types is clamped at a configurable value, such that no single type alone can lead to the exclusion of a document from the corpus. This was motivated by the fact that, for example, in many languages the personal pronoun for *I* is among the top ten types, but there are certain kinds of documents in which it does not occur at all because self-reference is sometimes considered inappropriate or unnecessary. The clamping value was set to 5 for all experiments described here. A short-document bias setting is also available, which reduces the Badness of short documents (because relative frequencies show a higher variance in short documents), but we currently do not use it in evaluations and in production runs.

3.2 Evaluation of Type Profiles

We use the ten most frequent types to generate frequency profiles, since the ten most frequent types usually make up for more than one fifth of the tokens in documents/corpora (Baroni, 2008), and they can be considered to have a reasonably domain-independent distribution. Figure 1 shows how the log-transformed weighted arithmetic mean and the corresponding standard deviation for the 10 most frequent types develop while training the DECOW2012 reference profile trained on 1,000 documents from the beginning of the crawl ("early profile"). As expected, both the mean and the standard deviation are relatively stable after 1,000 documents. The occasional jumps in the standard deviation (most remarkably for *und*) are caused by very long documents (sometimes over 1 MB of text) which thus receive very high weights. Future versions of the software will include a document size pre-filter and the option of using different profiles for documents of different length to smooth this out. However, given the evaluation results in Section 3.4, we think these additional mechanisms are not crucial.

3.3 Distribution of Badness Values

Next, we look at the distribution of the Badness values under realistic corpus processing scenarios. We used the early DECOW2012 profile described in Section 3.2 to calculate Badness values for a large number of early documents (2.2 GB HTML data; 27,468 documents), i. e., documents from the same phase of the crawl as the ones used for training the profile.[14] We did the same with early UKCOW2012 data (2.2 GB HTML data; 32,359 doc-

[12] In this sense, the method is, of course, not arbitrary, but based on quite reasonable theoretical assumptions about the distributions of words in texts.

[13] We have successfully used available n-gram-based language-identification in a task-specific crawling scenario (Barbaresi, 2013) and are planning to integrate all methods into one piece of software eventually.

[14] We use "early/late data" for "data from the early/late phase of the crawl" and "early/late profile" for "profile trained on a sample of the documents from the early/late phase of the crawl" from now on.

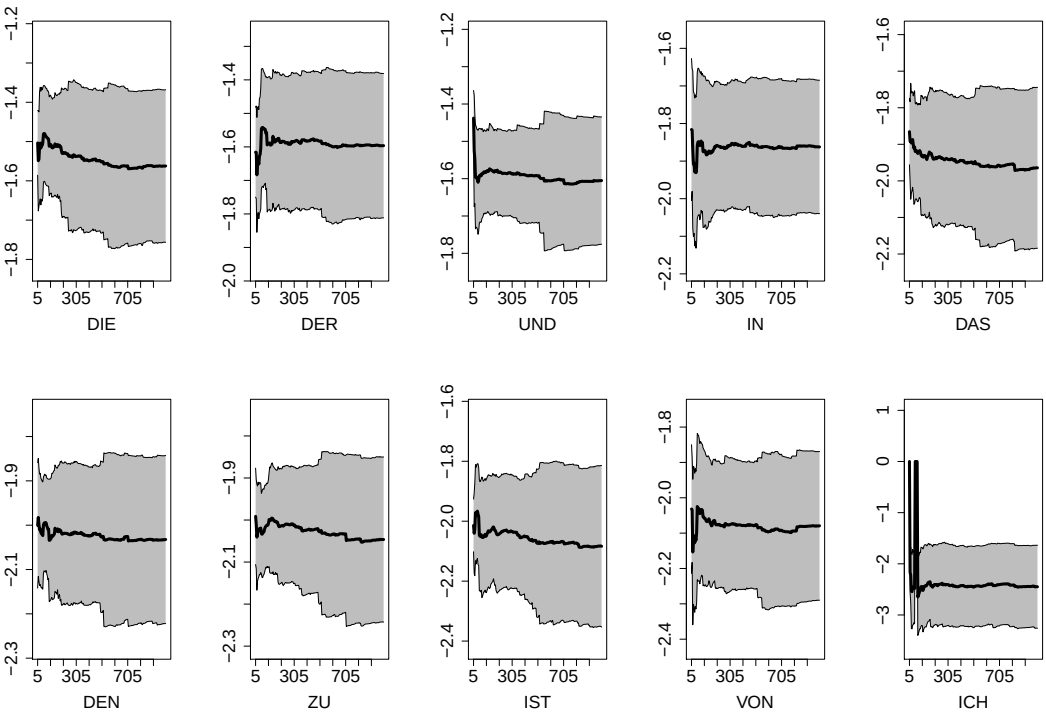

Figure 1: Development of the reference profile for DECOW2012 on early crawl data ("early profile") while training; x-axis: number of documents used for training; y-axis: log10-transformed weighted arithmetic mean of the respective type's frequency in the training documents; gray areas mark 1 standard deviation above and below the mean; the 10 most frequent types after 1,000 documents.

uments) and an early profile.[15] Figure 2 shows the resulting distribution of Badness values for documents above certain byte lengths.

In the early phase, the UKCOW2012 crawl found more short documents compared to the early phase of the DECOW2012 crawl, namely 4,891 (17,81%) documents more for 2.2 GB of raw data. The mean document length is therefore lower for UKCOW2012. This generally lower document length probably explains the different shape of the distribution, i. e., the higher overall Badness of the UKCOW2012 documents. In both cases, however, there are a lot of very bad documents (Badness=50) at short byte lengths. They are typically those documents which are completely or at least almost empty after boilerplate removal. For the following evaluations, we therefore removed all documents up to a length of 200 B.

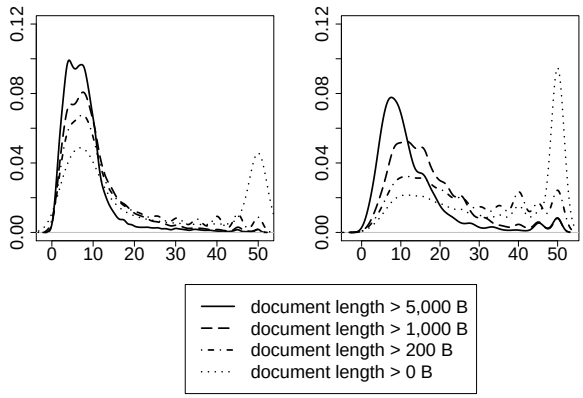

Figure 2: Density estimates for the distribution of Badness scores for the early profile on early data depending on document length; left: DECOW2012 ($n = 27,468$), right: UKCOW2012 ($n = 32,359$); x-axis: Badness score/threshold; y-axis: distribution density.

3.4 Comparison of Profiles

We now look at the question of whether profiles created from different samples have radically different effects. To this end, comparisons are made between the effects of profiles created *with* early and late data *on* early and late data, respectively.

[15] The UKCOW2012 early data here is a superset of the documents used in the coding task described in Section 2.

Figure 3 plots (for documents longer than 200 B) the proportion left over by early profiles on early data, etc.

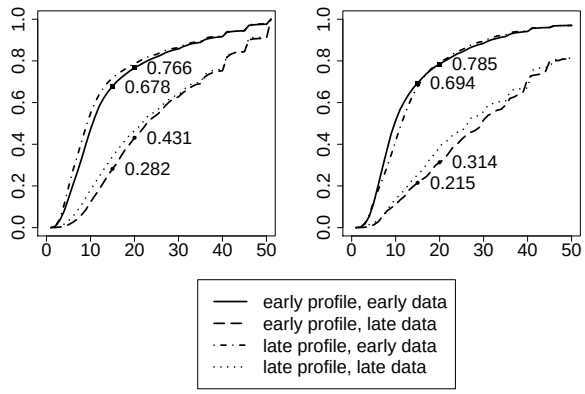

Figure 3: Effect of different profiles in terms of the proportion of documents left over at certain Badness thresholds ($\equiv$ cumulative density distribution of Badness values) for all documents longer than 200 B; left: DECOW2012 ($n = 27,468$ early; $n = 60,565$ late); right: UKCOW2012 ($n = 32,359$ early; $n = 34,879$ late); x-axis: Badness score/ threshold; y-axis: proportion of documents left in the corpus; values at Badness 15 and 20 for the early profile are given in the graphs.

In the case of DECOW2012, the early data sample contains documents which are on average 2.2 times longer than those in the late data sample. Profiles trained on documents from two such different samples would be likely candidates for having different effects. Surprisingly, the different profiles have rather negligible effects. For the early DECOW2012 data, the early profile leaves 76.6% of the document in the corpus, while the late profile leaves 78.6%, a difference of no more than 2%. On late data, it is 43.1% (early profile) and 46.4% (late profile). For the UKCOW2012 early data, it is 78.5% (early profile) vs. 79.2% (late profile) and for the late data 31.4% (early profile) and 39.1% (late profile). As expected, due to higher variance in the training data (which is mostly due to shorter document length), late profiles are more permissive, but the differences are not drastic. Figure 4 plots the raw agreement of the profiles on the early and the late data set at Badness thresholds from 1 to 50. It shows that the major difference is the reduced strictness of the late profiles on the early data, but mainly below thresholds of roughly 15 or lower.

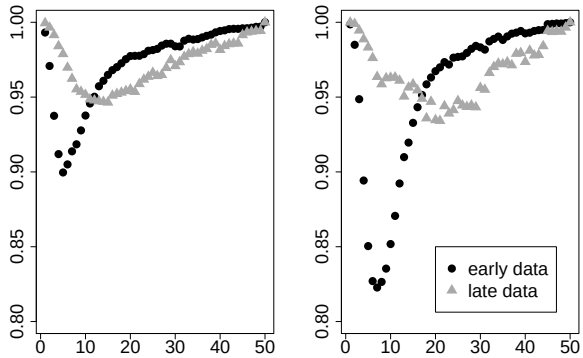

Figure 4: Profile comparison in terms of raw agreement between the profiles at thresholds [1..50]; left: DECOW2012; right: UKCOW2012; x-axis: thresholds; y-axis: proportion of identical decisions of the early and the late profile at the given threshold.

Finally, Figure 5 confirms the general picture. For the two TLDs (`.de` and `.uk`), it plots the Badness values calculated by the early and the late profiles on the two data sets. Each of the four plots corresponds to one data set (early or late) from one of the TLD crawls, and it compares the two profiles w. r. t. those data sets. Each dot represents a document, and it is positioned to show the Badness value assigned to that document by the late profile (x) and the early profile (y).

The linear models on the data show quite a strong correlation between the Badness scores assigned by the two profiles. The intercepts are higher for late data compared to earlier data (DECOW2012: early data 1.028, late data 2.194, UKCOW2012: early data 0.994, late data 3.741), showing again that the early profiles are more sensitive/strict than the late profiles.

Why the UKCOW2012 data is worse in general is impossible to ascertain. Since the seed URLs were collected in a similar way for both crawls, and the crawler software was configured in exactly identical ways, the difference is most likely a symptom of the unpredictable biases brought about by unselective Breadth-First Search.

3.5 Avoiding Impossible Decisions

So far, we have shown that deciding whether a document contains mostly text (as opposed to non-

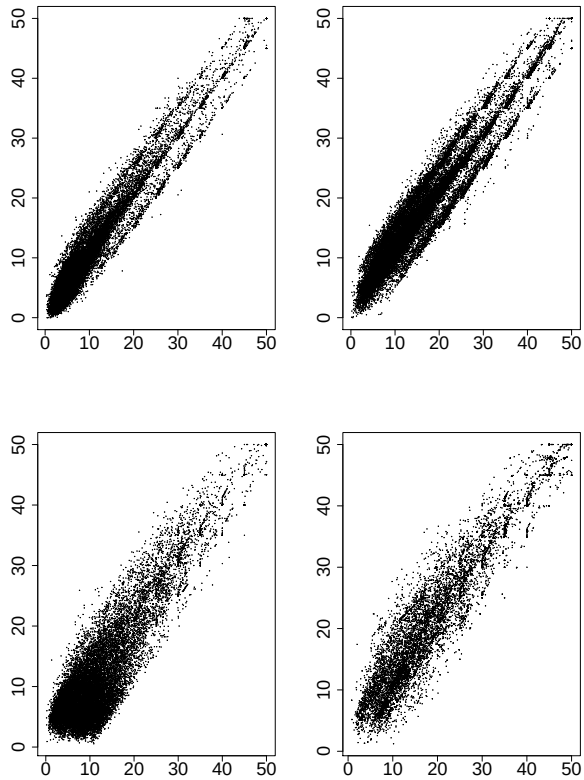

Figure 5: Comparison of profiles; top: DECOW2012; bottom: UKCOW2012; left: early data; right: late data; x-axis: late profile; y-axis: early profile; LM top left (DECOW2012 early data): intercept=1.028, coefficient=0.980, R^2=0.988; LM top right (DECOW2012 late data): intercept=2.194, coefficient=0.952, R^2=0.982; LM bottom left (UKCOW early data): intercept=0.128, coefficient=0.994, R^2=0.9701; LM bottom right (UKCOW late data): intercept=3.741, coefficient=0.930, R^2=0.970; artefacts around Badness increments of 5 result from the clamping (to 5) of the values which are added up to calculate Badness.

text material) is a task which leads to substantial disagreement between humans. Furthermore, we have argued that the lack of short and otherwise highly frequent words can be measured easily and with consistent results for the kind of data in which we are interested. However, if we want to use the Badness score as a document filter, then there still remains a threshold to be determined, i. e., a score above which documents are excluded from the corpus. We now discuss how such a value should

	prec	rec	F1	correct	baseline
S	0.914	0.959	0.936	0.888	0.849
A	0.856	0.973	0.911	0.851	0.781
R	0.808	0.976	0.884	0.811	0.738

Table 2: Performance of the Badness algorithm as a classifier evaluated against the human coder decisions; thresholds chosen to produce the maximal possible agreement with any coder (which is coder S): coder threshold 0; Badness threshold 35; raw agreement of the human coders is 0.831 at these settings (Fleiss' $\kappa = 0.660$).

be chosen by comparing the Badness scores for the 1,000 UKCOW2012 documents from the experiment described in Section 2 with the coders' decisions.

We searched for the best match between Badness scores and coder decisions and found that if we keep documents rated by coder S (the least strict coder) as 0 or better, then setting the Badness threshold to 35 results in a proportion of correct predictions of 0.888, cf. Table 2. This is the best achievable value for any coder and any Badness threshold with the data from our coding task.

For a (hypothetical) gold standard based on coder S, the Badness score method achieves a precision, a recall, and an F1 score of well over 0.9. Of course, since the baseline ("keep all documents") is quite high, this means an increase in accuracy of only 0.039 (roughly 4%) compared to the baseline. At the same time, Table 2 shows that at the optimal settings for coder S, the methods achieves a precision below 0.9 (more bad documents remaining in the corpus) relative to the decisions by the other coders. Still, even for the strictest coder (R), precision is above 0.8. The recall, however, is generally excellent.

We suggest that the best lesson to learn from these results is that corpus designers should not make too many destructive design decisions, ideally none at all. If we keep all documents accepted as good enough for corpus construction by the most tolerant coders, then all users can be sure that the material in which they are interested is still contained in the corpus (near-perfect recall for everyone). If, in addition to this, we annotate the documents with (ideally several) metrics like the Badness score, corpus users can decide to use more or less clean and/or good documents when making queries or generating statistics from

the corpus. In other words, corpus users should be put in a position to decide how important precision and recall are for their purposes. This is currently our general strategy, and we summarize it in more detail in the next and final section.

4 Achievements, Further Research, and Corpus Formats

As it was said in Section 1.2, our ultimate target in web corpus construction is the creation of highly representative samples from the population of web documents with the least possible error introduced through post processing and normalization. Therefore, in addition to working on improved crawling methods, we let users decide how strictly they want to filter potential noise. The measures which we have already implemented and used for the construction of the UKCOW2012 corpus (which, however, was still crawled using a Breadth-First Search) are the annotation of documents with Badness scores and the annotation of paragraphs with values indicating the likelihood that they are boilerplate. Since we are currently in the stage of evaluation of diverse methods, we still removed documents with a Badness of 15 or higher (not 35, as suggested in Section 3.5), and we removed paragraphs with a certain likelihood of being boilerplate (although not as strictly as in earlier COW2012 corpora). For COW2013, we are planning to keep all paragraphs and all documents below a Badness of 35.

Since we use the IMS Open Corpus Workbench (CWB) for corpus access, we needed to encode the Badness and boilerplate scores in a way such that they can be used in CQP queries.[16] Adding the raw numeric values to structural attributes is not a feasible way of doing this, because CWB would basically treat them as factors, not enabling queries restricted by arithmetic conditions on those values. In other words, querying for documents with a Badness smaller than r_1 and greater than r_2, etc., is impossible. We therefore encode the values as single alphabetic characters between *a* (best) to maximally *z* (worst). Badness values were encoded in increments of 2, such that $[0,2)$ is encoded as *a*, $[2,4)$ as *b*, etc. For example, restricting the search to documents with a Badness of 10 or better can be achieved by specifying the regular expression `[a-e]` for the Badness annotation layer.

[16] `http://cwb.sourceforge.net/`

Of course, the amount of data increases considerably with this highly non-destructive approach to post processing and normalization. From an empirical point of view, this simply is not a valid counter-argument. What is more, it is quite feasible to construct giga-token corpora in such a way on modern hardware without serious performance penalty, as we have demonstrated with UKCOW2012. Furthermore, given that uniform random sampling allows for smaller samples in order to achieve representativeness, the effect of non-destructive cleansing and normalization on corpus size can be compensated for in the long run by using smaller samples in the first place. While very huge (and traditionally cleaned/normalized) corpora in the region of several 10^7 tokens (Pomikálek et al., 2012) are surely very useful, for some applications in empirical linguistics, better is better, and bigger is not necessarily better.

Acknowledgments

We would like to thank Sarah Dietzfelbinger for her participation in the coding task. Also, we would like to thank three anonymous WaC 8 reviewers for their helpful comments. Felix Bildhauer's work was funded by the *Deutsche Forschungsgemeinschaft* through the *Sonderforschungsbereich* 632, project A6.

References

Adrien Barbaresi. 2013. Crawling microblogging services to gather language-classified URLs. Workflow and case study. In *Proceedings of ACL Student Research Workshop*, Sofia. To appear.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Marco Baroni. 2008. Distributions in text. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook*, pages 803–822. Walter de Gruyter, Berlin.

Petra Saskia Bayerl and Karsten Ingmar Paul. 2011. What determines inter-coder agreement in manual annotations? a meta-analytic investigation. *Computational Linguistics*, 37(4):699–725.

C. Biemann, G. Heyer, U. Quasthoff, and M. Richter. 2007. The Leipzig Corpora Collection - Monolingual corpora of standard size. In *Proceedings of Corpus Linguistics 2007*, Birmingham, UK.

Chris Biemann, Felix Bildhauer, Stefan Evert, Dirk Goldhahn, Uwe Quasthoff, Roland Schäfer, Johannes Simon, Leonard Swiezinski, and Torsten Zesch. 2013. Scalable construction of high-quality web corpora. *Special issue of JLCL*. In prep. The list of authors is preliminary and might reflect neither the order nor the actual list in the printed version.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Ted Dunning. 1994. Statistical identification of language. Technical Report MCCS-94-273, Computing Research Laboratory, New Mexico State University.

Gregory Grefenstette. 1995. Comparing two language identification schemes. In *Proceedings of the 3rd Internation conference on Statistical Analysis of Textual Data (JADT 1995)*, pages 263–268, Rome.

Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. 2000. On near-uniform URL sampling. In *Proceedings of the 9th International World Wide Web conference on Computer Networks: The International Journal of Computer and Telecommunications Networking*, pages 295–308. North-Holland Publishing Co.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to its Methodology*. Sage Publications, Beverly Hills.

Gordon Mohr, Michael Stack, Igor Ranitovic, Dan Avery, and Michele Kimpton. 2004. Introduction to Heritrix, an archival quality web crawler. In *Proceedings of the 4th International Web Archiving Workshop (IWAW'04)*.

Jan Pomikálek, Miloš Jakubíček, and Pavel Rychlý. 2012. Building a 70 billion word corpus of English from ClueWeb. In *Proceedings of LREC 08*, pages 502–506.

Paat Rusmevichientong, David M. Pennock, Steve Lawrence, and C. Lee Giles. 2001. Methods for sampling pages uniformly from the World Wide Web. In *In AAAI Fall Symposium on Using Uncertainty Within Computation*, pages 121–128.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul. ELRA.

Roland Schäfer and Felix Bildhauer. 2013. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, San Francisco.

Johannes Steger and Egon Stemle. 2005. Krdwrd architecture for unified processing of web content. In Iñaki Alegria, Igor Leturia, and Serge Sharoff, editors, *Proceedings of the Fifth Web as Corpus Workshop (WAC5)*, pages 63–70, San Sebastián. Elhuyar Fundazioa.

Radim Řehůřek and Milan Kolkus. 2009. Language identification on the web: Extending the dictionary method. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 5449 of *Lecture Notes in Computer Science*, pages 357–368. Springer Berlin Heidelberg.