Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6958–6966
Marseille, 11–16 May 2020

CCOHA: Clean Corpus of Historical American English

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, Sabine Schulte im Walde
Institute for Natural Language Processing
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
{alatrarm, schlecdk, jonas, schulte}@ims.uni-stuttgart.de

Abstract

Modelling language change is an increasingly important area of interest within the fields of sociolinguistics and historical linguistics. In recent years, there has been a growing number of publications whose main concern is studying changes that have occurred within the past centuries. The Corpus of Historical American English (COHA) is one of the most commonly used large corpora in diachronic studies in English. This paper describes methods applied to the downloadable version of the COHA corpus in order to overcome its main limitations, such as inconsistent lemmas and malformed tokens, without compromising its qualitative and distributional properties. The resulting clean corpus of historical American English (CCOHA) contains a larger number of cleaned word tokens which can offer better insights into language change and allow for a larger variety of tasks to be performed.

Keywords: COHA, Corpora, Historical Linguistics, Language Change

1. Introduction

Languages are in a constant process of evolution. That is, they constantly change over time on all levels of linguistic structure. These changes reflect—and are driven by—external factors such as cultural changes and technological advances (Blank, 1999; Fromkin et al., 2018). The field of historical or diachronic linguistics is concerned with the study and analysis of language change over time. Over the past two decades, researchers have shown an increased interest in the various aspects of diachronic language change. This can be attributed to the advances in technology such as the digitization of historical texts, improved computational power and availability of large-scale historical corpora designed specifically for diachronic studies (Tahmasebi et al., 2018; Tang, 2018; Bowern, 2019). Large historical corpora first appeared a decade ago and quickly gained popularity because they allow researchers to test hypotheses using computational approaches that are only possible with corpora of such volume (Kutuzov et al., 2018; Dubossarsky et al., 2019; Perrone et al., 2019; Schlechtweg et al., 2019).

The Corpus of Historical American English (COHA) (Davies, 2012) is a popular large-scale resource for studying lexical, syntactic and semantic change in English. Despite its many features and advantages, COHA is not without its limitations. These shortcomings, which include inconsistent lemmas and malformed tokens, can complicate certain tasks and increase the required time and effort to complete them. As a case in point, let us consider the original task for which we needed COHA. The task required sentence-level context extraction for a set of target words, but was hindered by the presence of malformed tokens around sentence boundaries. To clarify, let us consider Example 1. which shows two sentences that have been merged due to the boundary loss between the words in bold. When attempting to extract the sentence-level context for the target word *gay*, the following occurrence causes erroneous results such as the position of the target word within the sentence, and the length of the sentence.

(1) "[...] know many of the people. I have a daughter that's on the Sheriff's **Department.As** far as the gay issue, I don't give a damn one way or the other as long as they don't bother me."

In light of this, we explored the data in COHA with the intention of identifying limitations that may obstruct Natural Language Processing (NLP) tasks. Then we cleaned COHA as much as possible without compromising the qualitative and distributional properties of the original data. The remainder of this paper is structured as follows: In the next section, we describe the related work on data clean-up. Further, we give an overview of COHA and describe its features and limitations. Then, we discuss the approach taken to clean COHA and overcome its limitations in Section 4. The resulting clean corpus is presented and compared to the original corpus in Section 5.

2. Related Work

Data clean-up is an essential yet time consuming process in research (Hill and Hengchen, 2019). Over the years, there have been various attempts to clean corpora for both specific and general use in NLP with some contributions aiming to automate the process (Reynaert, 2006). In the field of machine translation, Imamura and Sumita (2002) present a method for cleaning bilingual corpora based on translation literality as measured by word-level and phrase-level correspondence in sentence pairs. As for more general applications, the special interest group of the Association for Computational Linguistics (ACL) on the Web as Corpus (ACL SIGWAC) released the shared task CLEANEVAL (Baroni et al., 2008), which aimed to clean web data for use as corpora in NLP. More recent efforts include Graën et al. (2014) who cleaned the Europarl Corpus, a collection of the European Parliament's debates. Similarly, Faaß and Eckart (2013) cleaned the German web corpus deWaC of the WaCky project (Baroni et al., 2009). Our work is close to that of Faaß and Eckart as we adopt a similar approach that requires several passes over the data with a measure to test the corpus quality.

3. COHA

The Corpus of Historical American English (COHA), developed by Brigham Young University, is a structured collection of carefully selected historical English texts taken from newspapers, popular magazines, fiction and non-fiction books published between 1810 and 2009. The corpus offers nearly 406 million words and around 107,000 texts. Additionally, it is balanced by genre, sub-genre and domain across decades. For example: the genre 'fiction' accounts for 48 to 55 percent of all texts in each decade starting with the 1810s and ending with the 2000s. The creators of COHA argue that this balance helps researchers ascertain that the changes they observe in COHA reflect 'real world' changes rather than artifacts of differences in genre balance (Davies, 2012).

While COHA can be searched for free using its web portal, there is a limit on the number of daily queries one can make.[1] Alternatively, the corpus can be purchased and downloaded in three different formats which we briefly describe here.

Database The first format is that of tabular data suitable for relational databases. This format contains three tables: (i) The 'lexicon' table which provides information about each word (including punctuation) in the corpus such as word form, lemma and POS tag. Every word is assigned a unique identifier using the 'wordID' field which is also the index or rank of the word. (ii) The 'sources' table which contains information about the text or document such as title, author, year of publication, number of words and genre. Each text is assigned a unique identifier using the 'textID' field. (iii) The 'corpus' table which connects the previous tables by mapping words to their texts. A typical row in this table shows only a wordID (taken from the lexicon) and the corresponding textID (taken from the sources table) to indicate which text the word appears in. Each row in this table is also assigned a unique identifier using the 'ID' field. The COHA web portal provides a brief description of the database along with illustrative sample data.[2]

Annotated Corpus The second format is tokenized data annotated for lemma and part-of-speech (POS) tags using CLAWS (Rayson and Garside, 1998). This is referred to as the tagged or annotated corpus format.

Linear Text Corpus The third format is linear text in paragraphs, which appears to have been generated from the tokens of the annotated corpus. All tokens, including punctuation, are separated by white space. This format is known as the text corpus.

To provide a better idea of these formats, we present a sample of the actual data from the file *fic_1813_7433.txt* in COHA. The database format is shown in Table 1 which depicts the mapping between the IDs of the first five words in the sentence and the text file ID. The annotated data format in Table 2 shows the tokens, lemmas and POS tags for the same words. The malformed token *&c.;* is present in all formats of the corpus.

[1] https://www.english-corpora.org/coha/

[2] https://www.corpusdata.org/database.asp/

Text ID	ID	Word ID
7433	47437489	474
7433	47437490	3
7433	47437491	244
7433	47437492	3301
7433	47437493	1

Table 1: Sample data from the downloadable version of COHA showing the database format.

Token	Lemma	POS
By	by	ii
the	the	at
same	same	da
rule	rule	nn1
,	,	y

Table 2: Sample data from the downloadable version of COHA showing the annotated data format.

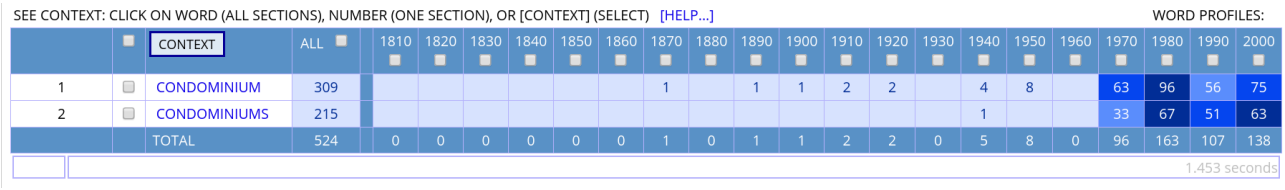
> By the same rule , is assigned to Summer the placid lake , **&c.;** not because that image is never seen [...] derived from a knowledge of its temperature .

Sample data from the downloadable version of COHA showing the linear text format.

An important aspect of the downloadable version of the corpus is that both the database format, via its lexicon table, and the linear text format stem from the annotated format of the corpus. According to the creators of COHA, the annotated data was created first, before the database which utilized not only the annotated tokens, but also their frequency and meta-data such as source document, year, and author (Davies, 2012, p. 125). This helped the creators of COHA manually correct errors for both formats. The last format created was the linear text which was generated using the tokens of the annotated corpus. The main drawback of this process is error propagation; errors not corrected in the annotated data will spread to the other formats and may to lead to more errors like incorrect frequency (database format) or loss of sentence boundaries (text format).

3.1. Features

At the time of its release in 2010, the structured nature of the data in COHA allowed it to provide researchers with useful features that were not available in larger unstructured corpora such as Google Books Ngrams (Google, 2010). The most common features of the COHA web portal include: word search, frequency charts, collocations, and key word in context (KWIC). Relevant to this paper is the word search feature, shown in Figure 1, which allows users to find occurrences of a target word within COHA using the word form or its morphosyntax. Figure 1 illustrates the results of running a search for the target word *condominium* as a noun. The word search feature is used during the evaluation process, which is presented in section 4.. A more comprehensive overview of the features of the web portal is provided by the creators of COHA (Davies, 2012).


SEE CONTEXT: CLICK ON WORD (ALL SECTIONS), NUMBER (ONE SECTION), OR [CONTEXT] (SELECT) [HELP...] WORD PROFILES:

		CONTEXT	ALL	1810	1820	1830	1840	1850	1860	1870	1880	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000
1		CONDOMINIUM	309							1		1	1	2	2		4	8		63	96	56	75
2		CONDOMINIUMS	215														1			33	67	51	63
		TOTAL	524	0	0	0	0	0	0	1	0	1	1	2	2	0	5	8	0	96	163	107	138

1.453 seconds

Figure 1: Search results for the noun *condominium* in COHA as obtained using the web portal.

3.2. Limitations

Despite offering various formats and useful features COHA is not without limitations. One known drawback of COHA is the lack of rare words which limits its use to studies of relatively common words (Tahmasebi et al., 2018). We briefly describe some of the other limitations we encountered while using the corpus.

Special Token '@' The documentation of COHA states that '@' tokens comprise 5% of the entire downloadable corpus due to legal reasons. In an effort to adhere to copyright regulations, the creators of COHA replace 10 consecutive tokens every 200 tokens with '@' characters for each text in the corpus. This replacement process prevents the use of these texts for their originally intended purpose as reading material.[3] However, this has several disadvantages: (i) Loss of tokens. (ii) For tasks where the context of a target word is needed, all instances containing '@' tokens will be discarded. (iii) Sentence boundaries can be lost as a result of the replacement process since '@' characters can replace punctuation. To illustrate, let us look at (2) which shows a sentence from the 1979 novel "Good as Gold" by Joseph Heller as it appears in the web portal results (2a) and in the downloadable corpus (2b). If we search for the target word *condominium*, (2b) can no longer be retrieved using this version of the corpus. Furthermore, the boundary between the sentences *What about your condominium?* and *His father was taken off guard* is lost.

(2) a. " Never mind my Niles, " he put it bluntly. " What about your condominium? " His father was taken off guard.

b. " Never mind my Niles , " he put it bluntly . " @ @ @ @ @ @ @ @ @ @ off guard .

Malformed Tokens The corpus contains malformed tokens which can be classified into three categories: (i) Malformed valid tokens that are combinations of valid words, punctuation, or other special characters. These tokens usually follow several patterns such as those in Table 3 where words are not separated from punctuation. (ii) Invalid tokens which contain punctuation or special characters and are not part of the original text. Most tokens in this category have the special string value "null" as their POS tag. (iii) Empty tokens containing the control character "NUL" which causes encoding errors. This control character is not to be confused with the special string "null" mentioned in the previous category as "NUL" is a single reserved character that signifies the end of a string in various programming languages. Subsequently, having this character as the token can lead to tokenization errors.

Malfomation Type	Examples
Valid malformed tokens	them:First there. But - - follows
Invalid malformed tokens	&c?; q! |p130
Empty tokens	Windows NUL character

Table 3: Examples of malformed tokens extracted from the downloadable text of COHA.

These malformed tokens are possibly artifacts of the digitization process which were not corrected, or artifacts of the data processing and clean-up which was performed using a web interface (Davies, 2012, cf.).

Malformed Lemmas Some of the lemmas in the corpus are malformed, and can be classified into three groups: (i) Malformed lemmas resulting from the malformed tokens. (ii) Malformed lemmas of valid tokens. (iii) Empty lemmas which contain only the control character "NUL". Notably, groups 2 and 3 have lemmas which contain special characters that cause encoding errors. As an example of the second group, we consider the lemma *sautée* which contains the french accent. This particular lemma is linked to valid well-formed tokens but causes encoding errors since the accented letter é seems to be corrupt in some files. The first row in Table 4 illustrates this case, as the token *sauteed* has the corrupt lemma *sautÃ©* instead of *sauté*.

Malformed POS Tags Malformed POS tags in COHA are those which contain only the control character "NUL". Unlike normal empty tags, malformed POS tags cause encoding errors.

Inconsistent Lemmas Another limitation is the fact that in some cases different lemmas exist for the same word forms. Again we consider the lemmas for various forms of the word *sautée*. As shown in Table 4 the lemma differences may be caused by diverse spellings. However, the different lemmas for the word *aesthetic* have forms with the same spelling. A final example where the lemma is not only different but also incorrect is the word *tape* where the lemmas *tape* and *tpe* both appear. This particular case could be an artifact of the manual correction process which occurred during the creation of the corpus.

[3] https://www.corpusdata.org/limitations.asp/

Token	Lemma	POS Tag
sauteed	sautÃⓒ	vv0
saute	sautÃⓒ	vv0_nn1
saut	sauté	vv0_nn1
sauteed	sauteed	nn1_vv0
sauteed	sauť	vv0
saute	saute	nn1
saut	saut	nn1
sauteing	sautÃⓒ	vvg
saut	NUL	vvi

Table 4: Various forms of the word *sautée* with different and at times malformed lemmas.

Escaped HTML Characters The last limitation in COHA affects the downloadable data and seems to originate from the process of preparing data for use in the web portal. Specifically, the downloaded data contains escaped hypertext markup language (HTML) characters which are automatically unescaped by browsers when using the COHA web portal. Moreover, some of these escaped characters are part of valid tokens and cannot be simply removed. Instances of this limitation include `MOIS&EACUTE`(MOIS&EACUTE) and `<center>`( ).

Formats All limitations mentioned here apply to both versions of the corpus: the web-accessible data and the downloadable corpus with its three formats. The only exceptions are the first limitation (@ tokens) and the last one (escaped HTML characters), both of which apply only to the downloadable corpus. Furthermore, it should be emphasized that the database format of the corpus excludes empty tokens, lemmas, and POS tags which leads to further loss of information. A final observation is that these limitations are present in both the annotated data format and the linear text format.

4. Cleaning Process

The effect of the above-described limitations is amplified when moving from studies on the word level to the sentence level. Such is the case for our original task where COHA was used to extract sentential context for a set of target words. The extracted context was to be composed of a triplet of sentences: the previous sentence, the current sentence containing the target word, and the following sentence. In order to determine sentence boundaries, we used a sentence tokenizer to acquire a list of sentences. Then, using these boundaries as a guideline, we attempted to rebuild sentences from the list of tokens in the annotated corpus. This was not possible for some sentences because the sentence tokenizer was able to split the malformed tokens with punctuation, which lead to a mismatch between the current sentence from the tokenizer and the current rebuilt sentence from the tokens list (which still contained the unsplit malformed tokens). To clarify, let us consider Example (3) which shows two different versions of the 95^{th} sentence in the annotated file "fic_2000_13995.txt". For this file, the sentence tokenizer produced sentence (3a) which ends with "do." since it was able to split the malformed token "do. I". On the other hand, the rebuilt sentence (3b), which was obtained by concatenating the tokens as they appeared in the tagged file, ends with the malformed token "do. I". Such malformed tokens cause mismatches when trying to reconstruct sentences from the annotated data since the sentence boundary is lost in the original annotated files. Moreover, we observed that it is not possible to use the database format or the linear text format instead since these formats were built from the annotated corpus and contain the same malformed tokens.

(3) a. And I did n't know what to do.
b. And I did n't know what to do. I

Both the database format, via its lexicon table, and the linear text format stem from the annotated format of the corpus. Keeping this in mind, we aimed to clean the annotated format first and then generate the dependent parts of the other formats using the cleaned corpus. Accordingly, the steps described in this section were performed on the annotated corpus.

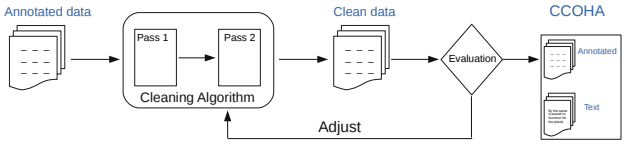


Figure 2: Diagram of the annotated corpus clean-up.

4.1. Annotated Corpus clean-up

The corpus clean-up was implemented using Python (Rossum, 1995) and the natural language toolkit (NLTK) (Bird and Loper, 2004). Specifically, the NLTK "Averaged Perceptron Tagger" was used to tag tokens, and NLTK "Punkt Sentence Tokenizer" was used to segment the data into sentences. The cleaning process, illustrated in Figure 2, was performed iteratively such that data were first cleaned and then manually evaluated. Based on the results of the evaluation, the cleaning algorithm would be updated and a new iteration would start where the original annotated corpus is cleaned and then evaluated. The cycle is repeated until the results of the evaluation reveal that no further improvements are needed. We explain the clean-up process in this subsection and explain the evaluation, which is based on our original task, in subsection 4.1.3.
In its final version, the cleaning script did two passes over the data. In the first pass, empty and "null" token and POS tags were cleaned, HTML characters were unescaped, and lemmas were unified for different forms of the same word. In the second pass, empty and "null" lemmas were cleaned, sentence boundaries were marked, and malformed tokens around sentence boundaries were cleaned (see Example 1.). We describe the clean-up process in more details in the following subsections.

4.1.1. First Pass

During this pass over the annotated data, all occurrences of the 'NUL' control character in the token form and POS tag fields were replaced with the special string "<nul>".

Malformed Token			First Pass			Second Pass		
Form	**Lemma**	**POS**	**Form**	**Lemma**	**POS**	**Form**	**Lemma**	**POS**
stripes–she	stripes–she	nn1	stripes	<temp>	<temp>	stripes	stripe	vv0_<sub>
			–	<temp>	<temp>	–	–	z_<sub>
			She	<temp>	<temp>	She	she	pphs1_<sub>

Table 5: Example of a malformed token before and after the cleaning process.

To detect this control character, we decoded the data from Windows-1252 then encoded it using UTF-8 and looked for the characters \x00, \00, and \0, which are incompatible with UTF-8 encoding. The lemma fields where the values were NUL were left for the second pass because contextual information from the surrounding tokens is required to correctly lemmatize any given token. The next step was to remove tokens where both the POS tag was equal to the special string "null" and the form was a non-word. Tokens that match these criteria include "<p>", "<>", and various control characters.

The following step was to unescape HTML characters. Next, the lemmas were unified for the different forms of the word *sautée* since some of them were corrupt and caused errors. This resulted in the unified lemma *saute* for the forms: *sauteed*, *sauted*, *saut*, *saute*, *sauteing*, *sautes*, and *sauting*. The final step of this pass aimed to identify malformed tokens away from sentence boundaries and when possible, split them into several valid tokens with the special string "<temp>" as the value for the lemma and POS tag. The special string "<temp>" reflects the temporary status of these fields as they were correctly filled during the second pass where contextual data from the entire sentence was available. Table 5 shows an illustrative example of this process. Clearly, the ambiguous word *stripes* can either be a verb or a noun and given the absence of contextual information during the first pass, it is not possible to lemmatize and tag this word with confidence. In the second pass however, both tagger and lemmatizer are able to correctly handle this word due to having the complete sentence-level context.

4.1.2. Second Pass

In this pass, the data from the previous pass was read and split into sentences using NLTK Punkt sentence tokenizer. Next, all occurrences of the 'NUL' control character in the lemma field were replaced with the special string "<nul>". Then, all tokens away from sentence boundaries where the lemma was either "<nul>" or "<temp>" were tagged and lemmatized given the full sentence as context. The only exception was the special token "@" which has a "<nul>" lemma. Similarly all tokens where the POS tag was "<nul>" were tagged and lemmatized in the same fashion. Considering that the NLTK "Averaged Perceptron Tagger" uses the Penn Treebank tagset (Marcus et al., 1994), the resulting POS tags were mapped to their CLAWS7 counterparts and appended with the special string "_<sub>" to help identify cleaned tokens. The mapping was manually created by the first author of this paper.

In order to detect the malformed tokens around sentence boundaries, sentences were reconstructed using the NLTK segmentation results as a guide. Specifically, upon reading each token in the annotated file, it would be appended to a list of tokens that were not part of the previous NLTK sentence. This list or "partial sentence" was then compared to the current NLTK sentence and when the sentences matched, a special end-of-sentence token ('<eos>) was added to the data to clearly mark the sentence boundary. Whenever the partial sentence was longer than the NLTK-based sentence, then the last added token, which is the current token being processed, was considered a malformed token and cleaned accordingly. The cleaning process for malformed tokens around sentence boundaries includes not only splitting, tagging and lemmatizing the new tokens, but also completing the sentence in order to match the NLTK-based sentence boundaries and then inserting the special end-of-sentence token.

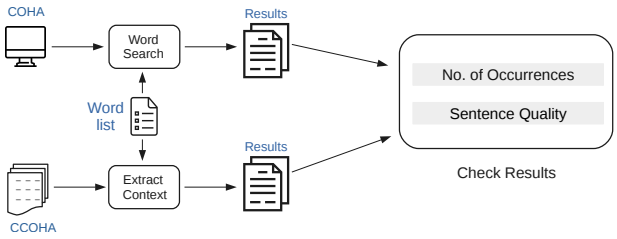


Figure 3: A Diagram of the evaluation process.

4.1.3. Evaluation

To prevent erroneous cleaning of valid tokens and ensure the maximum amount of limitations were overcome, the cleaned data were evaluated after every clean-up. The evaluation process is based on our original task which motivated the clean-up. To reiterate, The task required us to extract sentence-level context for a set of target words. The context consists of a triplet of sentences: the previous sentence, the current sentence containing the target word, and the next sentence. The set of target words contains 50 words.[4] The evaluation process,shown in Figure 3, consists of several steps. As can be seen, sentential context is extracted for the target words from both corpora: COHA, via the word search feature of its web portal, and CCOHA, by means of running a script to extract the contexts from the annotated data. The next step is to examine the quality of the extracted sentences and to compare the number of occurrences per target word in each corpus. That is to say, we compare the number of occurrence of the target word in CCOHA to that in COHA. Given the first limita-

[4] https://www.ims.uni-stuttgart.de/data/ccoha.

	COHA	CCOHA
Word Tokens	406,232,024	431,391,376
Non-Word Tokens	66,186,836	64,101,011
All Tokens	**472,418,860**	**495,492,387**
Lemma Types	795,806	2,246,898
Encoding	Windows-1252	UTF-8
Sentence Marker	None	<eos>
Available formats	Annotated, text, and database	Annotated and text

Table 6: Statistics for COHA before and after cleaning.

tion of the downloadable corpus, where tokens are replaced by '@' characters, we did not aim to match the number of retrieved contexts using the web portal, but rather aimed to increase the amount and quality of retrieved contexts in CCOHA. The quality was checked by manually inspecting the sentences to ensure the correctness of retrieved occurrences and to ensure that words and sentences contained minimal or no malformed and invalid tokens

We demonstrate this step by relying on one of the target words as a test case. Let us consider the noun *condominium* which occurs in the COHA corpus 524 times when searching using its lemma to ensure the inclusion of the plural form *condominiums*. When attempting to extract contexts for this lemma using the downloadable corpus we first obtained only 473 occurrences. Naturally, some cases were due to the replacement of tokens with the special symbol '@; (i.e. the first limitation). However, upon examining the results we observed some very large values for the in-sentence-position for some of the occurrences. Upon closer inspection, we noticed that the sentence boundaries were lost, which lead to the limitation of malformed tokens near sentence boundaries. Further qualitative examination revealed the other limitations such as HTML tags. Currently, the clean annotated corpus yields 498 results for the lemma *condominium*.

4.2. Cleaning Linear Texts

Acquiring a cleaned version of the linear text format of the corpus was a straightforward process. Namely, we used the cleaned annotated corpus to generate the linear text files for each document in the same format as the original linear text data. That is to say, all tokens were separated by white space including punctuation.

4.3. Cleaning the Database

Bearing in mind that our main task is not the cleansing of COHA but rather processing the annotated version to suit our needs, we were unable to spare the time and resources necessary to recreate the database files from the clean annotated corpus. This being said, it is possible to clean the database format by following these steps: (i) Rebuild the lexicon table to reflect the frequency and rank (wordID) changes. (ii) Update the corpus table to use the new updated word IDs.

5. Clean Corpus (CCOHA)

The resulting cleaned corpus CCOHA[5] uses UTF-8 character encoding and is larger than the original COHA corpus. The main differences shown in Table 6 reveal an increase of over 25 million word tokens and an increase of nearly two million non-word tokens such as dashes and end-of-sentence markers (<eos>). The large increase in the number of lemma types—nearly three times its original size—is indicative of the presence of new words in the clean corpus. However, it should be noted that part of this increase is attributed to the problem of inconsistent lemmas.

5.1. Features

Supplementary to the already existing features of COHA, this cleaned version provides some new useful features.

Sentence Boundary Markers Most sentences in the annotated corpus are now followed by a special token signaling the end of the sentence. As is shown in Table 7, the end-of-sentence token "_<eos>" has the same value for its lemma and POS tag to make it easier to identify and avoid erroneous inflation of the frequency of any POS tags.

No Empty Fields Currently, there are no more empty fields in the annotated corpus. All token forms or POS tags were initially filled with the special string "<nul>", then given valid values during clean-up. As for lemmas, a distinction must be made between the lemmas where the token form is the special replacement string '@' (first limitation) and those where the token form is something else (e.g., malformed or invalid). We observe a reduction of 3,562,464 in the number of "<nul>" lemmas where the token form is not equal to '@'.

An unintended limitation in the original corpus arises from the annotation of the special replacement tokens ('@'). More precisely, each '@' token is assigned a 'NUL' lemma and 'ii' POS tag which refers to general prepositions. Nevertheless, other tokens in the corpus have the same values for their lemmas and POS tags. In the downloadable COHA corpus, there are 14,402 such tokens. In contrast, the clean corpus CCOHA contains 10,881 of these tokens, which amounts to a 24.4% reduction. Although we believe this limitation can lead to inaccurate frequency counts when attempting to extract data using lemmas and POS tags without considering the token form, we did not assign a special lemma and POS tag to this token. The reasons for that are

[5]Find information on the availability of the corpus at `https://www.ims.uni-stuttgart.de/data/ccoha`.

the preservation the original data and the fact that the above problem can be resolved by considering the token form.

Token	Lemma	POS
He	he	pphs1
pictured	picture	vvd_<sub>
himself	himself	prp_<sub>
in	in	ii
sabots	sabot	nn2
and	and	cc
a	a	at1
rough	rough	jj
blue	blue	jj
peasant	peasant	nn1
smock	smock	nn1
.	.	y
<eos>	<eos>	<eos>

Table 7: A sentence from the clean annotated data in CCOHA.

Cleaned Fields Detection With regards to detection of modifications to the original corpus, the fact that POS tag fields for cleaned tokens end with the string "_<sub>" allows for convenient extraction of these tokens. Additionally, the mid-sized en dash and double dashes (−−) have been replaced by "<ndash>" in both tokens and lemma fields. These are part of the unescaped HTML characters which were causing encoding errors when using UTF-8 encoding. Extraction of cleaned fields may be needed for purposes of further processing, running a different lemmatizer or POS tagger, or for handling the HTML en dash character.

5.2. Limitations

Malformed Tokens As is the case with all automatic cleaning processes, the one presented in this paper missed a number of malformed tokens in the original corpus. Some of the types of malformed or invalid tokens that were missed are:

- Tokens that contain the pattern "P1X_1X_2" where X_1X_2 are digits in the range [0-9]. This pattern can come before or after the actual word in the token, and sometimes between two words. Furthermore, it is sometimes preceded by the |symbol. Instances of such tokens include "|p103And" and "Agnesp106said".
- Tokens where two or more words are not separated by white space. Examples of this are "sentimentalyarns", "endlesslyvariable" and "investigatingthose".
- Tokens tagged as "null" but are not control sequences or white space. The tokens "&Joni:wore?now;", "act" and "acts" are examples of such tokens.
- Malformed tokens containing numbers [0-9] and the special characters such as financial or mathematical ones (e.g., $+*%).

The special token "q!" which is present in the corpus was removed but its corresponding end-of-sentence marker (<eos>), which was added during clean-up, was deliberately kept because some files contain only the "q!" token. The decision to keep the end-of-sentence marker for such cases was motivated by the fact that removing this token meant these entries were now empty and should be deleted. However, the database format contained references to these files which would have been problematic when attempting to clean the database format.

Inconsistent Lemmas The limitation of inconsistent lemmas was not tackled during this clean-up process with the exception of the lemma *sautee* due to its malformed instances. Moreover, upon evaluating the results of the clean-up, we became aware that the NLTK WordNet lemmatizer produces lemmas that may differ from the ones already present in the original COHA corpus thereby contributing to this limitation. As a test case, let us examine the noun *aesthetics* which was part of the target words used for evaluation and error analysis. The only lemma for this noun in the original corpus was *aesthetics*, yet after the clean-up, the new lemma *aesthetic* appears as well for this noun.

POS Tag Granularity As mentioned earlier, the POS tags produced by the NLTK tagger for cleaned malformed tokens were mapped to CLAWS7 tags. Granted that some coarse grained tags do not exist in the CLAWS7 tagset, we extended the tagset for COHA to accommodate these tags. It is important to remember here that the original COHA already extended its tagset by introducing the tags "y" for punctuation (.,?), "zz" for single letters of the alphabet (a,b,... etc.) and tokens containing dashes, "z" for double dashes (−−), and "null" for invalid tokens. The tagset extension is summarized in Table 8 where it is possible to see how the special tag "y" was extended to include more punctuation like quotation marks (‘’“”) and symbols like the dollar sign ($).

Tag	Meaning	
	COHA	**CCOHA**
PRP	N\A	Personal pronoun (I, you, he)
PNQ	N\A	Wh-pronoun (what, who) Wh-possessive (whose)
Y	Punctuation (.) Invalid tokens(<>, &apps)	Punctuation & Invalid tokens Symbols ($, #, +)

Table 8: Extensions and additions to the COHA POS tagset.

6. Conclusion

This paper presented our approach taken to clean the downloadable version of the COHA corpus. The resulting corpus CCOHA offers more word tokens, less non-words, and less invalid tokens than the original COHA. While the annotated and linear text formats are available in CCOHA, the database format should be generated by interested parties.
In conclusion, we discuss some of the possible improvements and steps that can be taken to further clean the corpus. First, malformed tokens that contain the pattern "P1X_1X_2" may be cleaned using regular expressions.

Second, malformed tokens that consist of one or more words could be cleaned using one of the many approaches for compound word splitting (Koehn and Knight, 2003; Norvig, 2009; Macherey et al., 2011). Third, if one wishes to use the more fine-grained POS tags of CLAWS7, it is feasible to extract tokens tagged using the coarse-grained tags and then retag them using CLAWS tagger or some heuristics. Last, by following the steps in Section 4.3. the database format of the clean corpus can be generated from the annotated data.

7. Acknowledgements

We especially thank Mark Davies for his comments during the cleaning process. We would also like to thank our reviewers for their insightful feedback. The first and second authors were supported by the CRETA center funded by the German Ministry for Education and Research (BMBF) during the conduct of this research. The second author was additionally supported by the Konrad Adenauer Foundation.

8. Bibliographical References

Baroni, M., Chantree, F., Kilgarriff, A., and Sharoff, S. (2008). Cleaneval: A Competition for Cleaning Web Pages. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco. European Language Resources Association.

Bird, S. and Loper, E. (2004). NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Blank, A. (1999). Why Do New Meanings occur? A Cognitive Typology of the Motivations for Lexical Semantic Change. In Andreas Blank et al., editors, *Historical Semantics and Cognition*, Cognitive Linguistics Research, pages 61–89. Mouton de Gruyter, Berlin / New York.

Bowern, C. (2019). Semantic Change and Semantic Stability: Variation is Key. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 48–55, Florence, Italy. Association for Computational Linguistics.

Dubossarsky, H., Hengchen, S., Tahmasebi, N., and Schlechtweg, D. (2019). Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Faaß, G. and Eckart, K. (2013). SdeWaC: A Corpus of Parsable Sentences from the Web. In *Language Processing and Knowledge in the Web*, pages 61–68. Springer.

Fromkin, V., Rodman, R., and Hyams, N. (2018). *An Introduction to Language*. Cengage Learning, 11th edition.

Graën, J., Batinić, D., and Volk, M. (2014). Cleaning the Europarl Corpus for Linguistic Applications. In Josef Ruppenhofer et al., editors, *Proceedings of the 12th Conference on Natural Language Processing*, pages 222–227, Hildesheim. Universitätsverlag Hildesheim.

Hill, M. J. and Hengchen, S. (2019). Quantifying the Impact of Dirty OCR on Historical Text Analysis: Eighteenth Century Collections Online as a Case Study. *Digital Scholarship in the Humanities*.

Imamura, K. and Sumita, E. (2002). Bilingual Corpus Cleaning Focusing on Translation Literality. In *Proceedings of the 7th International Conference on Spoken Language Processing*.

Koehn, P. and Knight, K. (2003). Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–193, Budapest, Hungary. Association for Computational Linguistics.

Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. (2018). Diachronic Word Embeddings and Semantic Shifts: A Survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Macherey, K., Dai, A. M., Talbot, D., Popat, A. C., and Och, F. (2011). Language-Independent Compound Splitting with Morphological Operations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1395–1404. Association for Computational Linguistics.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the Workshop on Human Language Technology*, pages 114–119. Association for Computational Linguistics.

Norvig, P. (2009). Natural Language Corpus Data. In Toby Segaran et al., editors, *Beautiful Data*, chapter 14, pages 219–242. O'Reilly Media.

Perrone, V., Palma, M., Hengchen, S., Vatri, A., Smith, J. Q., and McGillivray, B. (2019). GASC: Genre-Aware Semantic Change for Ancient Greek. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 56–66, Florence, Italy. Association for Computational Linguistics.

Rayson, P. and Garside, R. (1998). The CLAWS Web Tagger. *ICAME Journal*, 22:121–123.

Reynaert, M. (2006). Corpus-Induced Corpus Clean-Up. In *Proceedings of the 5th Conference on Language Resources and Evaluation*, pages 461–464, Genoa, Italy. European Language Resources Association.

Rossum, G. (1995). *Python Reference Manual*. CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands.

Schlechtweg, D., Hätty, A., del Tredici, M., and Schulte im Walde, S. (2019). A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Tahmasebi, N., Borin, L., and Jatowt, A. (2018). Survey

of Computational Approaches to Diachronic Conceptual Change. *arXiv preprint arXiv:1811.06278*.

Tang, X. (2018). A State-of-the-Art of Semantic Change Computation. *Natural Language Engineering*, 24(5):649–676.

9. Language Resource References

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Davies, Mark. (2012). *Expanding Horizons in Historical Linguistics with the 400-Million Word Corpus of Historical American English*. Edinburgh University Press.

Google. (2010). *Google Books Ngram Viewer*. Google, Google Books Ngram Datasets, 1.0.